Technical white paper

HP Enterprise Virtual Array Storage and VMware vSphere 4.x and 5.x configuration best practices

Table of contents

Executive summary 4
The challenges 4
Key concepts and features 4
ALUA compliance 5
Configuring EVA arrays 6
Using Command View EVA 7
Running Command View EVA within a VM 7
Using the Storage Module for vCenter 8
Array hardware configuration and cabling 10
Disk group provisioning 13
Formatted capacity 13
Sparing overhead and drive failure protection level 13
Application-specific considerations 14
Storage optimization requirements 14
Vdisk provisioning 16
iSCSI configuration 17
Controller connections 20
Mezz: LUN groups and iSCSI targets 20
Front-end GbE and FC ports 20
ESX connectivity to EVA iSCSI 20
Implementing multi-pathing in vSphere 4.x and 5.x 23
Best practices for I/O path policy selection 27
Configuring multi-pathing 27
Displaying the SATP list 29
Connecting to an active-active EVA array in vSphere 4.0, 4.1 and 5.x 30
Caveats for connecting to vSphere 4.1 and 5.x 31
Caveats for multi-pathing in vSphere 4.x/5.x 32
Upgrading EVA microcode 34

Overview of vSphere 4.x/5.x storage 34
Using VMFS 35
Using RDM 35
Comparing supported features 36
Implementing a naming convention 36
Sizing the vSphere cluster 38
Aligning partitions 38
Enhancing storage performance 39
Optimizing queue depth 39
Using adaptive queuing 39
Using the paravirtualized virtual SCSI driver 40
Monitoring EVA performance in order to balance throughput 41
Optimizing I/O size 42
VMware vStorage API for Array Integration (VAAI) 43
VAAI alignment considerations 44
VAAI transfer size consideration 44
VAAI and EVA proxy operations considerations 44
VAAI and Business Copy and Continuous Access 46
VAAI operation concurrency 46
VMware Space Reclamation and EVA 47
ESX UNMAP support history 47
UNMAP in ESXi 5.0Ux and ESXi 5.x 48
UNMAP in ESXi 5.5 49
UNMAP alignment, size and EVA host mode considerations 50
Summary of best practices 51
How can I best configure my storage? 51
Which is the best I/O path policy to use for my storage? 51
How do I simplify storage management, even in a complex environment with multiple storage systems? 52
How can I best monitor and tune the EVA array in order to optimize performance? 52
How do I maintain the availability of Command View EVA deployed in a VM? 53
Summary 53
Glossary 53
Appendix A: Using SSSU to configure the EVA 55
Appendix B: Miscellaneous scripts/commands 56
Changing the default PSP 56
Setting the I/O path policy and attributes 56
Configuring the disk SCSI timeout for Windows and Linux guests 56

Appendix C: Balancing I/O throughput between controllers 57
Appendix D: Caveat for data-in-place upgrades and Continuous Access EVA 61
Appendix E: Configuring VMDirectPath I/O for Command View EVA in a VM 62
Sample configuration 62
Configuring the vSphere host 64
Configuring the array 68
Configuring the VM 69
For more information 71

Executive summary

The HP Enterprise Virtual Array (EVA) storage[1] family has been designed for mid-range and enterprise customers with critical requirements to improve storage utilization and scalability. EVA arrays can fulfill application-specific demands for transactional I/O performance, while supporting easy capacity expansion, instantaneous replication, and simplified storage administration.

The combination of an EVA array, HP Command View EVA software and VMware vSphere 4.x and 5.x provides a comprehensive solution that can simplify management and maximize the performance of a vSphere infrastructure.

HP continues to develop and improve best practices for deploying HP EVA Storage arrays with vSphere 4.x and 5.x. This white paper describes a broad range of best practices for a Fibre Channel (FC) and iSCSI implementation; Fibre Channel over Ethernet (FCoE) implementation is outside the scope of this paper.

Target audience: vSphere and SAN administrators that are familiar with the vSphere infrastructure and virtual storage features, the EVA array family, and Command View EVA.

DISCLAIMER OF WARRANTY

This document may contain the following HP or other software: XML, CLI statements, scripts, parameter files. These are provided as a courtesy, free of charge, "AS-IS" by Hewlett-Packard Company ("HP"). HP shall have no obligation to maintain or support this software. **HP MAKES NO EXPRESS OR IMPLIED WARRANTY OF ANY KIND REGARDING THIS SOFTWARE INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NON-INFRINGEMENT. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT OR ANY OTHER LEGAL THEORY, IN CONNECTION WITH OR ARISING OUT OF THE FURNISHING, PERFORMANCE OR USE OF THIS SOFTWARE.**

The challenges

With vSphere 5, VMware continues to stretch the boundaries of scalability through features that include:

- Support for 1TB of RAM per virtual machine (VM)
- Support for 32 virtual CPUs per VM
- Support for 160 cores for host server
- High-performance SCSI virtual adapter
- Support for up to 512 VMs on a single ESX host[2]
- Modular storage stack that is closely integrated with the particular storage array

For administrators, this feature-packed hypervisor raises several questions about effectively configuring, tuning and deploying vSphere 4.x and 5.x in their respective SANs such as:

- What is the optimal storage configuration?
- Which I/O path policy is the most appropriate?
- How to simplify storage management, even in a complex environment with multiple storage systems?
- How to effectively monitor the SAN in order to quickly make adjustments when needed?

Successfully addressing these challenges is imperative if you wish to maximize the return on investment (ROI) for your SAN while continuing to meet the changing needs of the business. To help you achieve these goals, this paper presents best practices for configuring, tuning, and deploying a vSphere SAN environment.

Key concepts and features

This section introduces key concepts and features associated with the successful configuration, tuning, and deployment of a vSphere SAN environment. These include Asymmetric Logical Unit Access (ALUA) compliance, virtual disk (Vdisk) controller ownership and access, and Vdisk follow-over.

[1] All references in this document to HP EVA Storage arrays imply classic HP EVA Storage arrays and the rebranded HP EVA P6000 Storage.
[2] Depending on host server resources

ALUA compliance

All EVA storage solutions – models P6x00, EVA8x00/6x00/4x00 – are dual-controller asymmetric active-active arrays that are compliant with the SCSI ALUA standard for Vdisk access/failover and I/O processing.

Note
ALUA is part of the SCSI Primary Commands – 3 (SPC-3) standard.

While the active-active nature of the array allows I/O requests to a Vdisk to be serviced by either controller, the array's asymmetry forces the optimal access path to the Vdisk to be used (that is, the I/O path to the controller that requires less processing overhead).

The controller with the optimal path to the Vdisk – managing controller – can issue I/Os directly to the Vdisk, whereas the non-managing controller – proxy controller – can receive I/O requests but must pass them to the managing controller to initiate fulfillment.

The following example shows how a read I/O request sent to the non-managing controller is processed:

1. The non-managing controller transfers the request (proxy) to the managing controller for the Vdisk.
2. The managing controller issues the I/O request to the Vdisk and caches the resulting data.
3. The managing controller then transfers the data to the non-managing controller, allowing the request to be fulfilled via the controller/server ports through which the server initiated the request.

Thus, a proxy read – a read through the non-managing controller – generates processing overhead.

Note
Since write requests are automatically mirrored to both controllers' caches for enhanced fault tolerance, they are not affected by proxy processing overhead. Thus, the managing controller always has a copy of the write request in its local cache and can process the request without the need for a proxy from the non-managing controller.

Vdisk controller ownership and access

The ability to identify and alter Vdisk controller ownership is defined by the ALUA standard.

EVA arrays support the following ALUA modes:

- Implicit ALUA mode (implicit transition) – The array can assign and change the managing controller for the Vdisk
- Explicit ALUA mode (explicit transition) – A host driver can set or change the managing controller for the Vdisk

EVA arrays also support the following ALUA access types:

- Active-Optimized (AO) – The path to the Vdisk is through the managing controller
- Active-Non-Optimized (ANO) – The path to the Vdisk is through the non-managing controller

ALUA compliance in vSphere 4.x and 5.x

ALUA compliance was one of the major features added to the vSphere 4 SCSI architecture and remains standard in vSphere 4.1 and 5.x. The hypervisor can detect whether a storage system is ALUA-capable; if so, the hypervisor can optimize I/O processing and detect Vdisk failover between controllers.

vSphere 4.x and 5.x support all four ALUA modes:

- Not supported
- Implicit transitions
- Explicit transitions
- Both implicit and explicit transitions

In addition, vSphere 4.x supports all five ALUA access types:

- AO
- ANO

- Standby – The path to the Vdisk is inactive and must be activated before I/Os can be issued
- Unavailable – The path to the Vdisk is unavailable through this controller
- Transitioning – The Vdisk is transitioning between any two of the access types defined above

The following load-balancing I/O path policies are supported by vSphere 4.x and 5.x:

- Round Robin – ALUA-aware
- Most Recently Used (MRU) – ALUA-aware
- Fixed_AP – ALUA-aware (Introduced in ESX 4.1. Rolled into Fixed I/O in ESXi 5.x.)
- Fixed I/O – Not ALUA-aware

Because they are ALUA-aware, Round Robin and MRU I/O path policies first attempt to schedule I/O requests to a Vdisk through a path that includes the managing controller.

For more information, refer to Configuring multi-pathing.

Vdisk follow-over

Another important concept that must be understood is Vdisk follow-over, which is closely associated with ALUA.

As described above, ALUA defines which controller in an asymmetric active-active array is the managing controller for a Vdisk. In addition, follow-over ensures that, when the optimal path to the Vdisk changes, all hosts accessing the Vdisk change their access paths to the Vdisk accordingly.

Follow-over capability is critical in a vSphere 4.x and 5.x cluster, ensuring that Vdisk thrashing[3] between controllers cannot occur. With follow-over, all vSphere servers[4] accessing a particular Vdisk update their optimal Vdisk access paths accordingly when the Vdisk is implicitly moved from one controller to the other.

Configuring EVA arrays

HP provides tools to help you configure and maintain EVA arrays. For example, intuitive Command View EVA can be used to simplify day-to-day storage administration, allowing you to create or delete Vdisks, create data replication groups, monitor the health of system components, and much more. For batch operations, HP recommends using Storage System Scripting Utility (SSSU), a command-line tool that can help you quickly deploy large EVA configurations, back them up for future deployments, and perform advanced administrative tasks.

When configuring a large number of Vdisks for a vSphere 4.x and 5.x implementation, such as that described in this paper, you should configure the Vdisks to alternate between EVA controllers using either Path A-Failover/failback or Path B-Failover/failback (see Vdisk provisioning).

Appendix A: Using SSSU to configure the EVA presents a sample script that creates multiple Vdisks, alternates path preferences between two controllers, and presents the Vdisks to vSphere servers.

This section outlines the following options for configuring an EVA array:

- Using Command View EVA
- Running Command View EVA within a VM
- Using the HP Insight Control Storage Module for vCenter

[3] Backward and forward transitioning
[4] Also known as ESX servers

Using Command View EVA

Command View EVA can manage an array using one of the following methods:

- Server-based management (SBM) – Command View EVA is deployed on a standalone server that has access to the EVA storage being managed.

  If desired, you can run Command View EVA within a VM, allowing you to clone this VM and use it as a template that can be quickly redeployed as needed, with minimal reconfiguration required.

  When running Command View EVA in a VM, the following modes are supported:

  – Virtualized SCSI mode – The Command View EVA instance can only view the active path to LUN 0 on a single controller. This mode is enabled by default when you install Command View EVA 9.2 or later.

  – VMDirectPath mode – The Command View EVA instance bypasses the virtualization layer and can directly manage the EVA storage. This mode is supported on Command View EVA 9.3 or later.
- Array-based management (ABM) – For supported EVA models, Command View EVA can be deployed on the array's management module.

Running Command View EVA within a VM

Your ability to deploy Command View EVA within a VM may be impacted by the following:

- EVA model
- Command View EVA version being used
- Availability of a host bus adapter (HBA) that can be dedicated to the VM

Table 1 compares the options for deploying Command View EVA in a VM.

Table 1. Comparing requirements for Virtualized SCSI and VMDirectPath modes

Requirement	Virtualized SCSI	VMDirectPath
Minimum EVA firmware	XCS 09534000 or later	All
Minimum Command View EVA software	9.2 or later	9.3 or later
Dedicated HBA required	No	Yes
Compatible EVA models	EVA8400 EVA6400 EVA4400 EVA4400-S P6300 P6500	EVA4100 EVA6100 EVA8100 EVA8400 EVA6400 EVA4400 EVA4400-S P6300 P6500
Command View EVAPerf support	No	Yes
SSSU support	Yes	Yes
VMware vMotion support	Yes	No

Caveats for installing Command View EVA in a VM

If you are installing Command View EVA in a VM, take care where you deploy the VMware virtual disk hosting the operating system on which you plan to install Command View EVA. If you were to deploy this virtual disk on the same array that you intend to manage with the Command View EVA instance, any issue that causes the EVA array to be inaccessible would also impact your ability to identify this issue.

In general, when deploying Command View EVA in a VM, use a virtual disk on the vSphere host's local datastore, if available.

If you must deploy the Command View EVA VM on the SAN, then consider deploying two or more instances on two or more storage systems to increase availability in the event of an array failure.

Best practices for deploying Command View EVA in a VM with VMDirectPath I/O

- Deploy Command View EVA on the local datastore of the particular vSphere server.
- If a SAN-based Command View EVA deployment is required, then deploy instances on multiple arrays within the infrastructure to ensure the management interface remains available in the event the array hosting the VM for the primary Command View EVA instance becomes inaccessible.

For information on configuring VMDirectPath for use with Command View EVA, refer to Appendix E: Configuring VMDirectPath I/O for Command View EVA in a VM. For supported hardware and software components please refer to VMware's KB for Command View EVA at http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1025641

Using the Storage Module for vCenter

HP Insight Control for VMware vCenter Server seamlessly integrates the manageability features of HP ProLiant, HP BladeSystem, HP Virtual Connect, and HP Storage inside VMware vCenter Server. Administrators gain insight and control of their HP infrastructure supporting their VMware virtualized environment from a single pane of glass – reducing the time it takes to make important decisions, increase capacity, and manage planned and unplanned downtime. The single download includes three modules:

- Recovery Manager for VMware (RMV): Manages recovery sets of HP 3PAR StoreServ storage systems
- Server module: Licensed with HP Insight Control and manages HP ProLiant servers, HP BladeSystem, and HP Virtual Connect.
- Storage module: Free to use with HP Storage and leverages deep capabilities of HP's portfolio of storage arrays while providing a consistent view, regardless of the HP storage type. The following HP Storage families of storage systems are supported (see the HP Single Point of Connectivity Knowledge (SPOCK) website (HP Passport credentials are needed to access the site) for the complete list):
  - HP 3PAR StoreServ
  - HP StoreVirtual Storage and StoreVirtual VSA
  - HP EVA
  - HP XP
  - HP MSA
  - HP StoreOnce Backup and StoreOnce VSA

  Use the Storage module to:

  - Monitor the status and health of HP storage systems
  - View detailed storage configuration including thin-provisioning, replications, and paths data
  - View relationships between the virtual and physical environment
  - Provision datastores and virtual machines easily:
    - Create, delete and expand datastores
    - Create new virtual machines onto new datastores in one easy step
    - Clone virtual machines using array-based snapshot and snap clone technologies
    - Delete orphan volumes

  Figure 1 shows a storage summary screen for a host that provides a quick, at-a-glance view of HP storage systems and datastores.

Figure 1. Storage summary screen

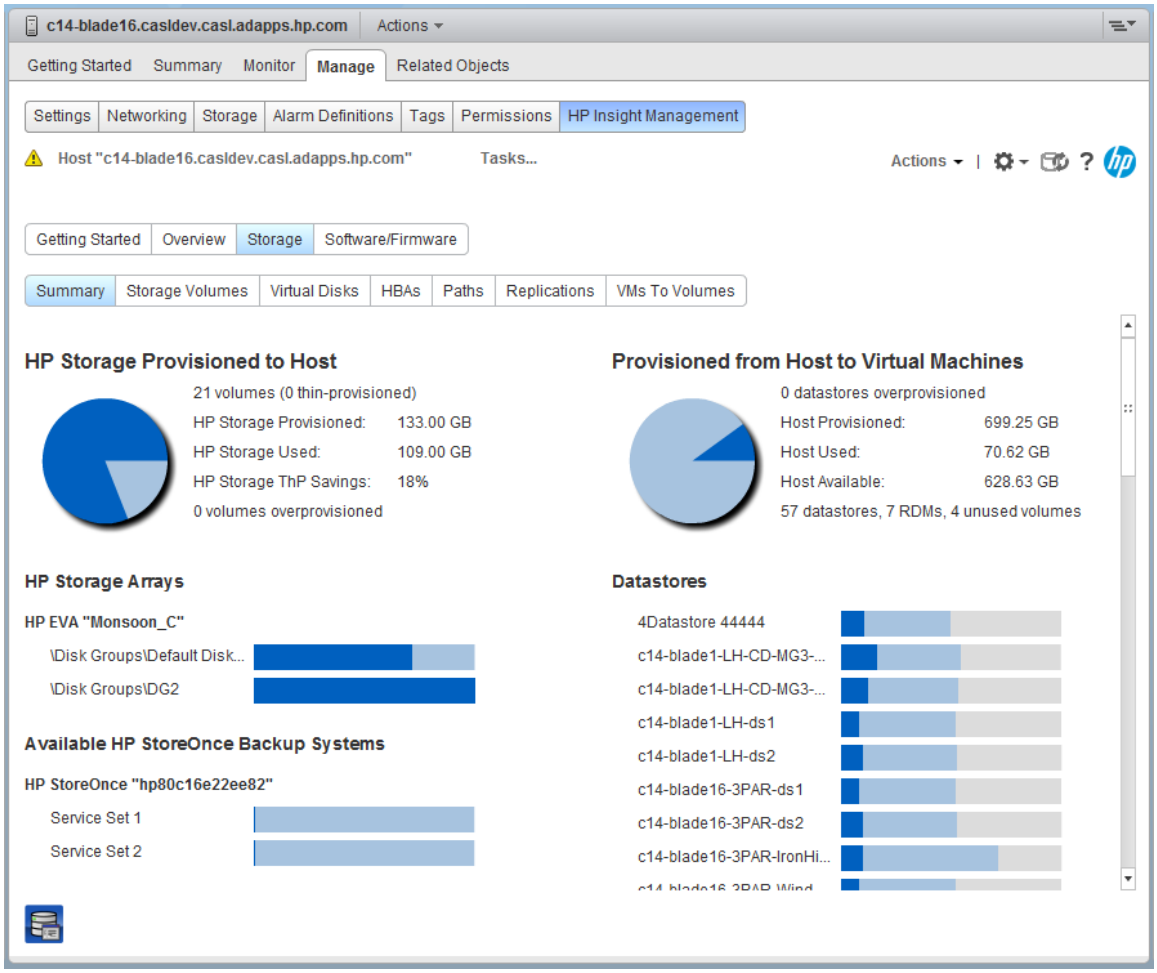
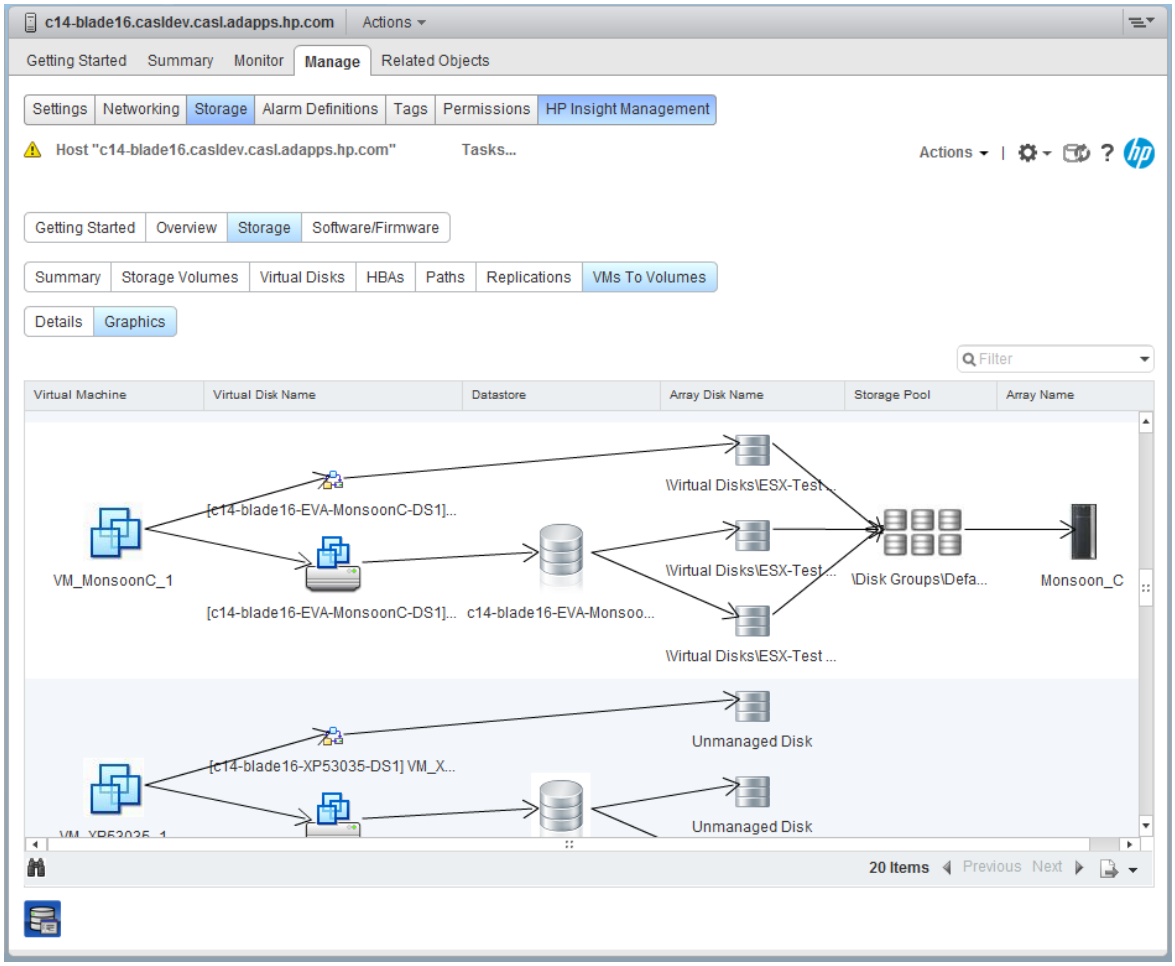


Figure 2. The Storage module of the plug-in provides mapping from the virtual to the physical environment.



The Storage module enhances VMware functionality by detailing the relationships between the virtual and physical environment. For example, Figure 2 shows the mapping from the virtual machine to the array on which it resides. Understanding the physical infrastructure is crucial for making well-informed decisions when designing, deploying, maintaining, and troubleshooting a virtual environment.

Best practice for mapping virtual objects to storage and for monitoring and provisioning storage

- Use the Storage Module for vCenter to save time and improve efficiency by mapping, monitoring, provisioning, and troubleshooting EVA storage directly from vCenter.

For more information on the Storage Module for vCenter, refer to the HP Insight Control for VMware vCenter Server User Guide.

Array hardware configuration and cabling

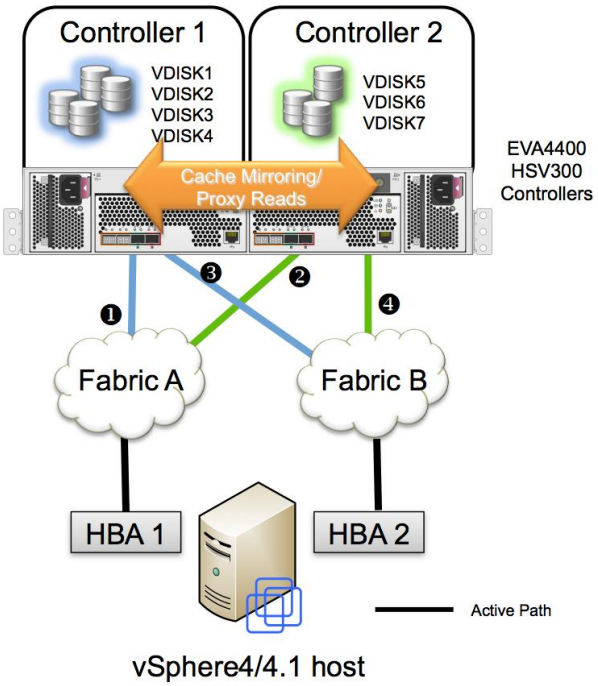
Best practices for EVA hardware configuration and cabling are well defined in the HP 6400/8400 Enterprise Virtual Array user guide and, thus, are not described in great detail in this paper.

However, it is important to note that when configuring vSphere to access an EVA array, HP highly recommends creating a redundant SAN environment by leveraging the following components:

- Redundant controllers on the EVA array
- Redundant Fibre Channel SANs
- At a minimum, dual HBAs in each ESX host

The resulting topology should be similar to that presented in Figure 3, which shows a vSphere 4.x server attached to an EVA4400 array through a redundant fabric.

Figure 3. Highly-available EVA/vSphere 4.x SAN topology



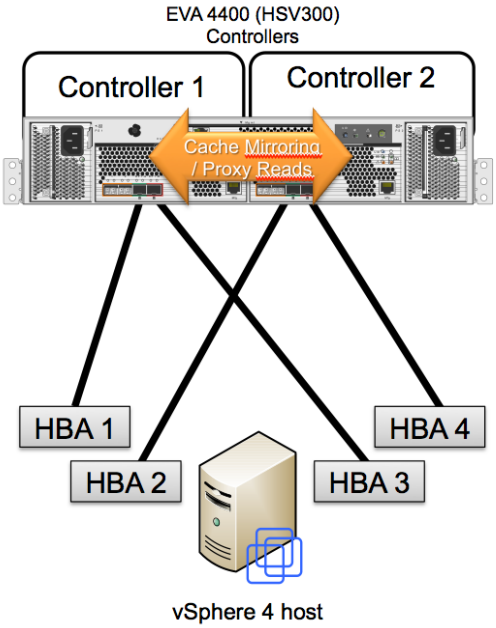
The benefits of this topology include the following:

- The topology provides increased fault tolerance, with protection against the failure of an HBA, a single fabric, or a controller port or controller.
- As described earlier, it is a best practice to access the Vdisk through the managing (optimal) controller for read I/O. Thus, in the event of an HBA failure, failover occurs at the HBA-level, allowing the Vdisk to remain on the same controller.

  For this reason, HP highly recommends configuring each HBA port in the ESX server to access one or more ports on each controller.
- Similarly, a controller failure only triggers a controller failover and does not cause an HBA failover, reducing system recovery time in the event of a failure.
- A fabric failure does not trigger a controller failover or force the use of the proxy path.

In a direct-connect environment, the same principles can be achieved with two more HBA or HBA ports; however, the configuration is slightly different, as shown in Figure 4.

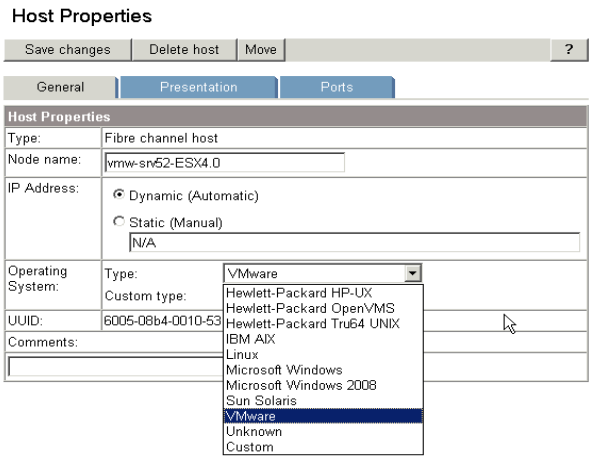
Figure 4. EVA/vSphere 4.x and 5.x direct-connect topology



If the direct-connect configuration were to use two rather than four HBA ports, there would be a one-to-one relationship between every HBA and controller. Thus, a controller failover would result in an HBA failover and vice versa, creating a configuration that is not ideal.

In order to implement the recommended topology shown in Figure 3, all vSphere hosts must have their EVA host profiles set to **VMware** in Command View EVA, as shown in Figure 5.

Figure 5. EVA host profile for a vSphere host



Note

When configuring VMware Consolidated Backup (VCB) with an EVA array, all vSphere hosts must be set to **VMware**. However, the VCB proxy host, which is a Microsoft® Windows® server attached to the EVA, must be set to **Microsoft Windows** (Windows Server 2003) or **Microsoft Windows 2008** (Windows Server 2008).

Disk group provisioning

An EVA disk group is the largest storage object within the EVA storage virtualization scheme and is made up of a minimum of eight physical disks for FC, SAS or FATA drives and 6 for SSD drives. Within a disk group, you can create logical units of various sizes and RAID levels.

Notes

- An EVA RAID level is referred to as VraidX.
- The EVA array allows Vraid0, Vraid1, Vraid5, and Vraid6 logical units to coexist in the same disk group on the same physical spindles.
- Not all EVA models support all disk drive types mentioned above.
- SSD drives require their own disk group.
- FATA drives require their own disk group.
- FC drivers require their own disk group.
- SAS drives require their own disk group.

When configuring an EVA disk group for vSphere 4.x and 5.x, keep in mind the following important factors:

- Formatted capacity
- Sparing overhead and drive failure protection level
- Application being virtualized
- Storage optimization requirements

Formatted capacity

Although disk capacity is typically referred to in round, integer numbers, actual storage capacity in an EVA is tracked in binary form; thus, the formatted capacity of a drive may be slightly smaller than its nominal capacity. For example, a drive with a nominal capacity of 146 GB provides 136.73 GB of formatted capacity, approximately 6.5% less.

Sparing overhead and drive failure protection level

Sparing overhead in an EVA array is defined as the amount of space that must be reserved to be able to recover from physical disk failures within a disk group. Unlike other storage systems, which set aside specific disks as spare drives, the EVA spreads the spare space[5] across all the disks within a group. The EVA sparing implementation eliminates the possibility of an assigned spare disk failing when it is needed for recovery.

The overhead created by sparing is calculated as follows:

$$Sparing\ capacity\ =\ (Size\ of\ largest\ disk\ in\ disk\ group\ *\ 2)\ *\ (Protection\ level)$$

In this formula, the value for disk drive failure protection level (Protection level) may be as follows:

- None
- Single – The disk group survives the failure of a single disk
- Double – The disk group survives the failure of two disks

[5] Also known as reconstruction space

You can use Command View EVA to set a disk drive failure protection level in the properties for the particular disk group, as shown in Figure 6.

Figure 6. Disk protection level as seen in Command View EVA

Note
Vraid0 Vdisks are not protected.

For a disk group where the largest disk drive capacity is 146 GB and double disk drive failure protection is required, sparing capacity can be calculated as follows:

$$Sparing\ capacity\ =\ (136.73\ *\ 2)\ *\ 2\ =\ 546.92\ GB$$

Sparing does not span disk groups; each disk group must be allocated its own sparing space based on the above formula.

Application-specific considerations

One of the most common misconceptions about server virtualization is that when an application is virtualized, its storage requirement can be reduced or changed. In practice, due to the aggregation of resources, virtualization typically increases the storage requirement. Thus, when you virtualize an application, you should maintain the storage required by this application while also provisioning additional storage for the virtual infrastructure running the application.

Sizing storage for any application that is being virtualized begins with understanding the characteristics of the workload. The following formula is used to calculate the spindle count required for a random access workload:

$$Number\ of\ drives\ needed\ =\ \frac{(Total\ IOPS\ *\ RAID\ penalty\ *\ Write\%)\ +\ (Total\ IOPS\ *\ Read\%)}{Raw\ performance\ of\ the\ disk\ drive}$$

In this formula, the **Total IOPS** value and read/write ratio are application-dependent.

The **RAID penalty** value is defined as the number of I/Os to disk that result from a guest I/O due to the particular Vraid level being used. For example, every I/O request to a Vraid1 Vdisk results in two I/Os being issued in the array in order to provide data protection.

Best practice for sizing an EVA disk group

- When sizing an EVA disk group, start by determining the characteristics of the application's workload, which will help you optimize array performance.

Storage optimization requirements

In addition to the number of disks required to handle the performance characteristics of the particular application, you must also account for the total storage capacity required by the applications and VMs being deployed.

This storage capacity can be determined by simple arithmetic by adding the storage requirements for each VM to the capacity required for the various applications. However, depending on your particular storage optimization objective, the actual formatted capacity yielded can be lower than the simple aggregation of the required number of EVA drives.

HP defines three storage optimization schemes, each of which is subject to specific storage overhead and deployment considerations:

- Cost
- Availability
- Performance

Optimizing for cost

When optimizing for cost, you goal is to minimize the cost per GB (or MB). Thus, it makes sense to minimize the number of disk groups; in addition, since the cost per GB is lower as drive capacity increases, it is best to use the largest disks of the same capacity within a particular disk group.

Even if you have disks with different capacities, it is better to use them in a single disk group rather than creating multiple disk groups.

Best practice for filling the EVA array

- To optimize performance, fill the EVA with as many disks as possible using the largest, equal-capacity disks. Note that the use a few larger drives with many small drives is inefficient due to sparing considerations.

Optimizing for availability

When optimizing for availability, your goal is to accommodate particular levels of failures in the array.

Availability within the array – and its usable capacity – can be impacted by a range of factors, including:

- Disk group type (enhanced or basic)
- Vdisk Vraid type
- Number and variety of disks in the disk group
- Protection levels
- Use of array-based copies

Typically, the additional protection provided by using double disk drive failure protection at the disk group-level cannot be justified given the capacity implications; indeed, single disk drive failure protection is generally adequate for most environments.

Enhanced disk groups offer features such as:

- Vraid6 protection, along with the traditional EVA Vraid0, 1, and 5
- Additional metadata protection, which is associated with a further storage capacity overhead

Many database applications use Vraid1 for database log files to guarantee performance and availability. However, while providing stronger data protection than Vraid5, Vraid1 has a much higher storage cost.

Best practice for protecting the disk group

- Single disk drive protection is sufficient for a disk group unless the mean time to repair (MTTR) is longer than seven days.

Best practices for using Vraid

- Only use Vraid6 when it is a requirement for your deployment.
- Vraid5 comes at a lower storage-capacity cost and provides adequate redundancy for most ESX deployments.

Optimizing for performance

When optimizing for performance, your goal is to drive as much performance as possible from the system.

However, configuring for optimal performance may have an impact on usable storage capacity. For example, it may be desirable to segregate small random workloads with short response time requirements from sequential workloads. In this use case, you should create two disk groups, even though this configuration would create additional sparing capacity utilization and reserved sparing capacity within each disk group.

Best practice for using disks of various performance characteristics

- When using disks of varying performance characteristics, use a single disk group rather than multiple disk groups.

Summary

- The use of a single EVA disk group is typically adequate for all storage optimization types (cost, performance, capacity).

Vdisk provisioning

All EVA active-active arrays are asymmetrical and comply with the SCSI ALUA standard.

When creating a Vdisk on an EVA array, you have the following options for specifying the preference for the managing controller for that Vdisk:

- No Preference
  - Controller ownership is non-deterministic; Vdisk ownership alternates between controllers during initial presentation or when controllers are restarted.
  - On controller failover (owning controller fails), Vdisks are owned by the surviving controller.
  - On controller failback (previous owning controller returns), Vdisks remain on the surviving controller; no failback occurs unless explicitly triggered.
- Path A-Failover only
  - At presentation, the Vdisk is owned by Controller A.
  - On controller failover, the Vdisk is owned by Controller B.
  - On controller failback, the Vdisk remains on Controller B; no failback occurs unless explicitly triggered.
- Path A-Failover/failback
  - At presentation, the Vdisk is owned by Controller A.
  - On controller failover, the Vdisk is owned by Controller B.
  - On controller failback, the Vdisk is owned by Controller A.
- Path B-Failover only
  - At presentation, the Vdisk is owned by Controller B.
  - On controller failover, the Vdisk is owned by Controller A.
  - On controller failback, the Vdisk is owned by Controller A; no failback occurs unless explicitly triggered.
- Path B-Failover/failback
  - At presentation, the Vdisk is owned by Controller B.
  - On controller failover, the Vdisk is owned by Controller A.
  - On controller failback, the Vdisk is owned by Controller B.

In the event of a controller failure that triggers the failover of all owned Vdisks to the alternate controller, it is critical that, when the failed controller is restored, Vdisk ownerships that were failed over to the surviving controller are failed back to the restored controller. This action ensures that, after a controller failover, the system can return to its default balanced/configured state and that all Vdisks do not end up on a single controller, degrading system performance.

With vSphere 4.x and 5.x and earlier versions of ESX, it is highly recommended to use one of the following preferences when configuring EVA Vdisks:

- Path A-Failover/failback
- Path B-Failover/failback

These preferences ensure that Vdisk ownership is restored to the appropriate controller when a failed controller is restored.

Vdisks should be created with their controller failover/failback preference alternating between Controller A and B.

The above recommendations provide additional benefits in a multi-pathing configuration, as described in Configuring multi-pathing.

Best practice controller ownership

- Controller ownership for EVA Vdisks should alternate between Controller A and Controller B, using the Path-A-Failover/Failback or Path-B-Failover/Failback setting.

iSCSI configuration

The HP EVA family of arrays offers a variety of iSCSI connectivity options. Depending on the array, iSCSI connectivity can be achieved through the iSCSI option built into the array controller (HP EVA P63x0/P65x0) or through the use of the HP MPX200 Multifunction Router.

Figures 7, 8 and 9 outline the three configuration options.

Figure 7. HP EVA P6000 with 1GbE iSCSI Module

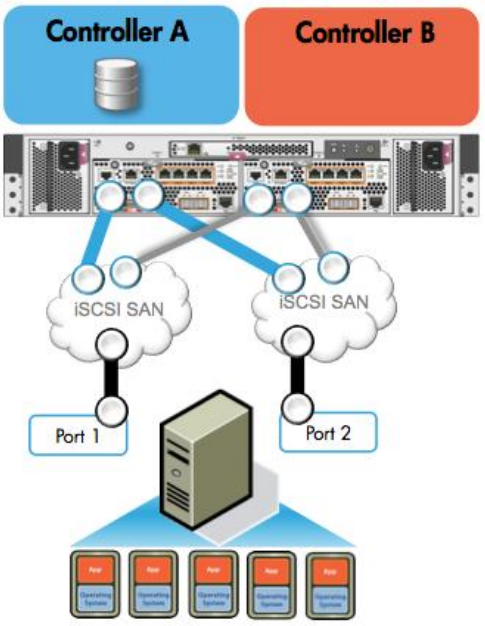


Figure 8. HP EVA P6000 with 10GbE iSCSI/FCoE Module

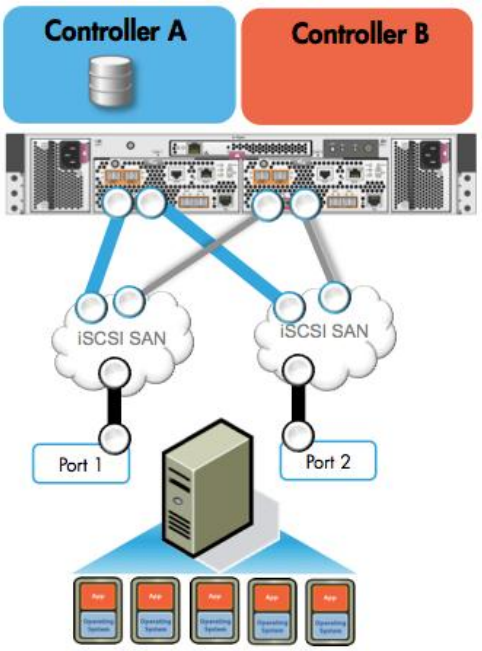
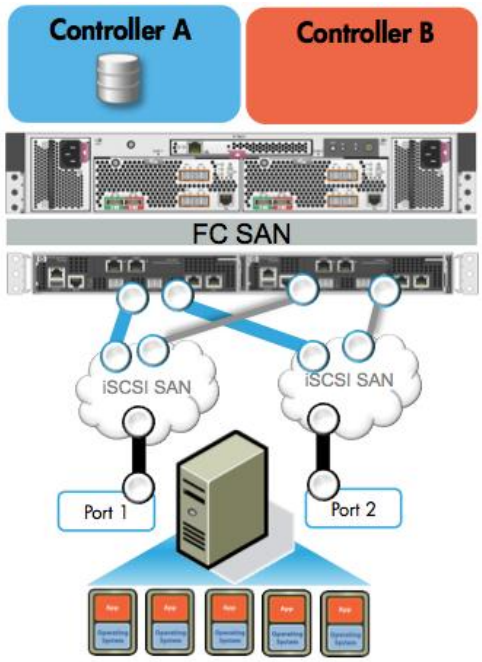


Figure 9. HP EVA with MPX200 Multifunction Router



The HP EVA P6000 1GbE iSCSI Module more commonly referred to as the 1GbE iSCSI Module as shown in Figure 7 provides each EVA controller with 4 1GbE ports or a total of 8 iSCSI ports per array.

The HP EVA P6000 10GbE iSCSI Module as shown in Figure 8 enables 2 10GbE iSCSI ports on each controller or a total of 4 iSCSI ports per array.

The MPX200 Multifunction Router enables a wide range of connectivity options. The MPX200 offers 2 built-in 1GbE ports, and an additional 2 1GbE ports with the optional 1GbE blade or 2 10GbE ports with the optional 10GbE blade. Additionally the MPX200 has 2 8Gb Fibre Channel ports used for FC SAN connectivity to the FC array(s) that the MPX200 is providing iSCSI connectivity for. An HP EVA connects to the MPX200 via Fibre Channel fabric and exports LUNs for iSCSI access through the MPX front-end iSCSI ports. Up to 4 EVAs can connect in this fashion to a single MPX200.

As shown in Figure 9, MPX200 configurations are typically deployed with a minimum of 2 MPX200 Multifunction Routers for redundancy.

In order to efficiently and appropriately attach a vSphere SAN via iSCSI to HP EVA Storage it is critical to understand the inner workings of the MPX200 and 1GbE iSCSI Module and more specifically how they expose iSCSI targets to hosts for access.

Figures 10 and 11 show high-level architecture diagrams of the internals of the HP EVA with 1GbE iSCSI Module and 10GbE iSCSI Module options. The basic architecture is comprised of three layers:

- EVA Controllers
- Mezz: LUN groups and iSCSI targets
- Front end GbE and FC ports

Figure 10. HP EVA with 1GbE iSCSI Module option – Architecture diagram

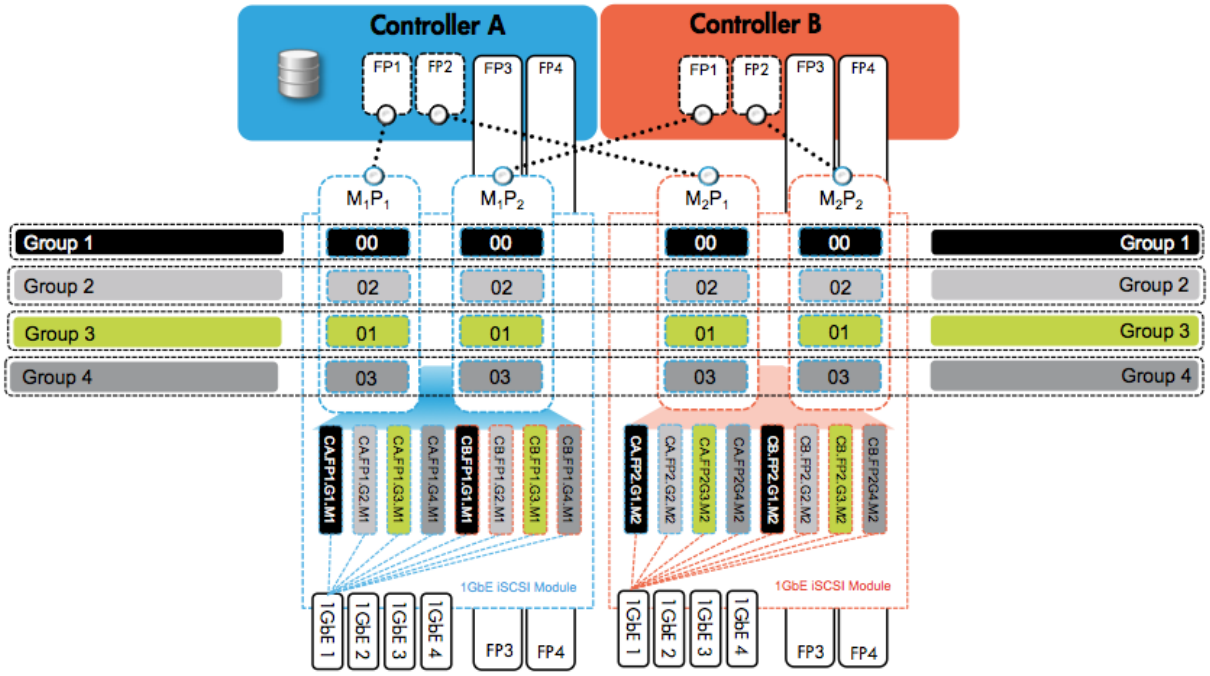
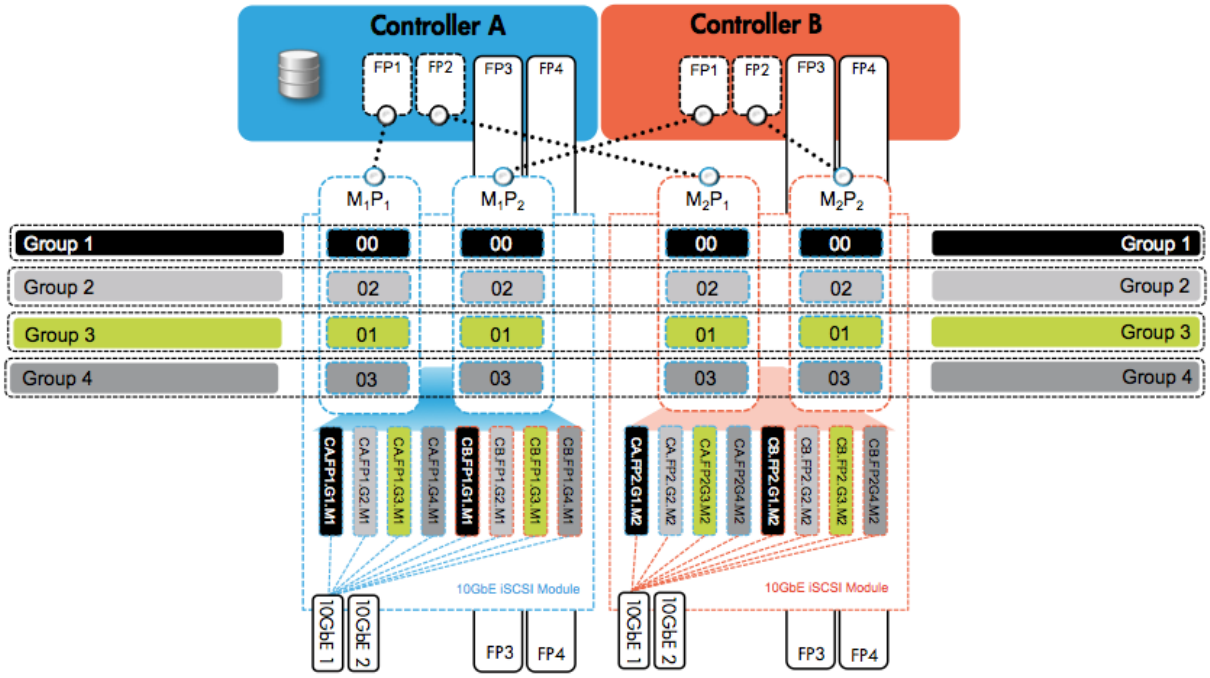


Figure 11. HP EVA with 10GbE iSCSI Module option – Architecture diagram



Controller connections

As shown in Figures 10 and 11, Fibre Channel ports FP1 and FP2 on each controller are connected to the 1GbE iSCSI Module and 10GbE iSCSI/FCoE Module respectively. For redundancy each array controller has a connection to the 1GbE iSCSI Module available in the other controller.

- Controller A FP1 connects to Mezz 1 Port 1 (M_1P_1)
- Controller B FP1 connects to Mezz1 Port 2 (M_1P_2)
- Controller A FP2 connects to Mezz 2 Port 1(M_2P_1)
- Controller B FP2 connects to Mezz 2 Port 2 (M_2P_2)

The controllers to Mezz connections are all Fibre Channel internal connections. This is similar to connecting an EVA to an MPX200 using physical front-end Fibre Channel ports, except that with the Mezz option, the connections are internal to the controller.

Mezz: LUN groups and iSCSI targets

Within the Mezz, LUN groups are defined. These groups can be seen as virtual buckets that hold LUNs for iSCSI presentation to hosts. By design, the array defines four LUN groups and each LUN group has a theoretical limit of 1024 LUNs; however, in practice it can support up to 256 LUNs. Each group also spans the iSCSI Module in both controllers. With the use of N-Port Virtualization (NPIV), each LUN group has four initiator connections to the EVA controllers; one connection through each of the Mezz ports M_1P_1, M_1P_2, M_2P_1, and M_2P_2. In Figures 10 and 11, the port numbers 00, 01, 02, 03 below each Mezz port, represents these NPIV initiator connections. Since the P6000/EVA allows up to 256 LUNs per initiator each of the NPIV port numbers in each group could theoretically access 256 individual LUNs. However for redundancy reasons all four NPIV ports in a iSCSI Module group must expose access to the same LUNs. For this, the maximum number of addressable LUNs per group is 256.

Each group has 4 NPIV connections to the EVA controllers; two connections to each EVA controller; which yields two iSCSI targets per group for each iSCSI Module. A total of 8 iSCSI targets are created in this fashion per iSCSI Module and all 8 targets are mapped for host access through each GbE port on the iSCSI Module.

When a LUN is created via Command View EVA for presentation to a host through iSCSI, it is created and assigned to one of the available LUN groups. This assignment is important because it has a direct relation to the iSCSI targets the LUN will be accessible through by the host. In Figure 10, assuming the LUN shown on Controller A was created in Group 1 (Group highlighted in black), then this LUN would only be accessible via each GbE port through the iSCSI targets:

- CA-FP1-**G1**-M1: ControllerA, FP1, **Group1**, Mezz1
- CB-FP1-**G1**-M1: ControllerB, FP1, **Group1**, Mezz1
- CA-FP2-**G1**-M2: ControllerA, FP2, **Group1**, Mezz2
- CB-FP2-**G1**-M2: ControllerB, FP2, **Group1**, Mezz2

All the other iSCSI targets shown in the Figure 10/11 will not expose this LUN to hosts because those iSCSI targets do not connect to Group 1.

Front-end GbE and FC ports

The 8 iSCSI targets on each controller built from the iSCSI Module connections are mapped to each GbE port. For example, on the 1GbE iSCSI Module, each of the 8 1GbE ports per iSCSI Module will allow access to all iSCSI targets. Similarly, on the 10GbE iSCSI Module, the 8 iSCSI targets per iSCSI Module are mapped to each of the two 10GbE ports on each iSCSI Module.

ESX connectivity to EVA iSCSI

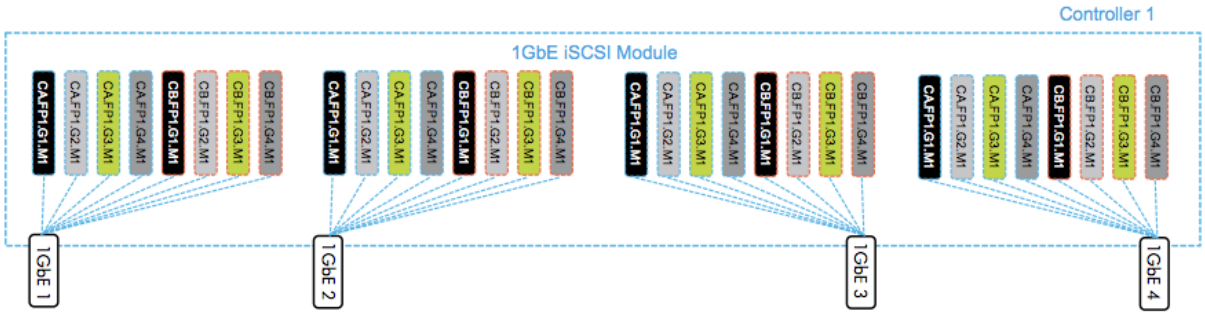
Availability consideration

ESX 4.x and 5.x allow up to eight iSCSI paths per LUN. When connected to an EVA with the 1GbE iSCSI Module option careful configuration steps must be taken to avoid the possibility of exhausting all eight iSCSI paths per LUN without having the proper high availability in the environment.

As ESX detects paths to LUNs during LUN discovery, it is possible that ESX will discover all eight iSCSI targets hence paths to a LUN through controller 1 before discovering any paths to LUNs through controller 2. Figure 12 below, illustrates this logical connection and assumes that the same physical connections are made to controller 2.

Figure 12 below, shows the iSCSI target to GbE port connections from a single 1GbE iSCSI Module perspective on one array controller.

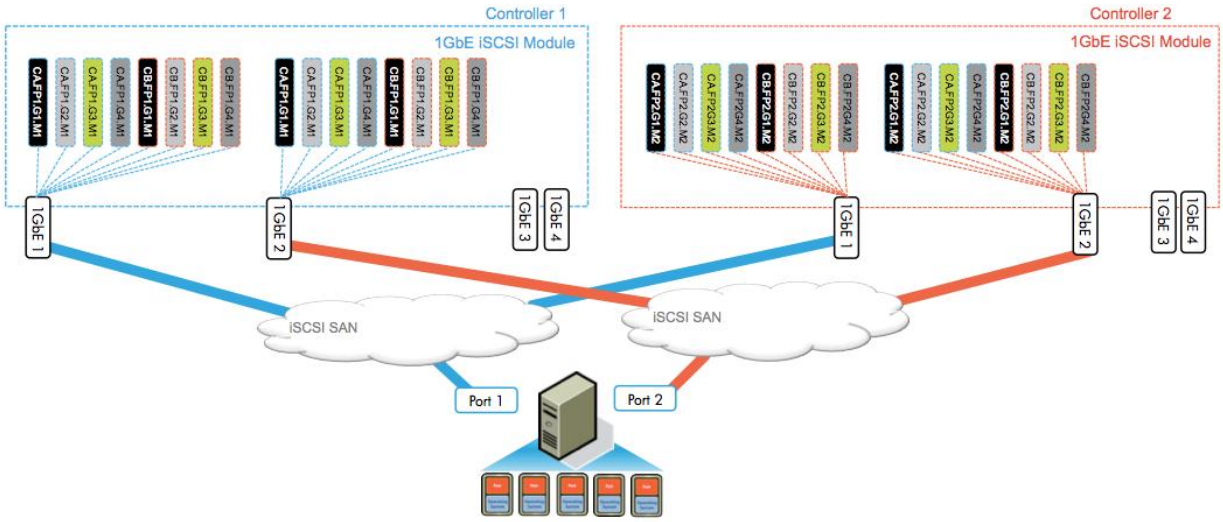
Figure 12. Logical view of iSCSI target connection to GbE ports – Single controller view.



From an ESX host perspective, ESX has the proper NIC redundancy and will detect its maximum of eight paths per LUN. If NIC 1 (port 1 in Figure 12) failed, I/O would be failed over to an available iSCSI target through NIC 2 (port 2 in Figure 13). However, from an array perspective, all paths to LUNs are built through iSCSI targets on the same controller. If this controller was to fail or reboot, the host would lose all access to storage because the maximum number of paths was exhausted through access to controller 1 and no paths to LUNs were built using the paths through controller 2.

When using an EVA with a 1GbE iSCSI Module option and all GbE ports must be attached to the iSCSI SAN, it is recommended to use Static discovery on the ESX side and manually select which iSCSI targets will be exposed to ESX for LUN access. Alternatively, a more adequate high availability configuration is shown in Figure 14 below where only two GbE ports per controller are attached to the iSCSI SAN and two NICs are used in the host.

Figure 13. Logical view of iSCSI target connection to GbE ports – Dual controller view.



In this configuration, iSCSI Dynamic discovery can be used to enable ease of deployment while ensuring proper high availability.

In an EVA with a 1GbE iSCSI Module option configuration, high availability via redundant paths is dependent on the following factors:

- The number of NIC adapters in the host accessing the iSCSI SAN
- The number of iSCSI targets exposed to the iSCSI SAN
- The iSCSI discovery type used

Table 2 below provides guidance to select the right number of paths at the host and at the array to maintain proper high availability in a 1GbE iSCSI Module configuration with two distinct SANs as shown in Figure 10.

Table 2. HP EVA 1GbE iSCSI Module High Availability configuration decision chart

	Used 1GbE Port per controller	1		2		3		4	
	Used iSCSI targets used per 1GbE port for each LUN	1	2	1	2	1	2	1	2
Used NIC ports per host	1	☒	☒	☒	☒	☒	☒	☒	☒
	2	✓	✓	✓	✓	✓	x	✓	x
	3	✓	✓	✓	x	x	x	x	x
	4	✓	✓	✓	x	x	x	x	x

☒ - Does not meet adequate high availability

✓- Meet adequate path count and high availability

x - Exceeds the ESX supported Max path per LUN – Does not meet adequate or symmetrical high availability

Table 2 is derived from the following facts:

- The 1GbE iSCSI Module has four ports per controller.
- A LUN that is exposed via iSCSI using EVA iSCSI Modules belongs to one and only of the four LUN groups.
- Each LUN group exposes its LUNs through two iSCSI targets and both iSCSI targets are accessible through all GbE controller ports.
- A user can use one or all available GbE ports when configuring vSphere 4.x/5.x with HP EVA 1GbE iSCSI Module.
- A user can elect to only use one of the two iSCSI targets to access the LUNs within a LUN group.

HP EVA iSCSI connectivity provides a wide range of configuration options. Each option comes with a tradeoff between increased path availability and increased GbE port bandwidth. In Table 2, the highlighted cell is the intersection of configuration options that provides the best balance of high availability and GbE port bandwidth.

Best practices for 1GbE iSCSI Module configuration

- When all GbE ports are connected to the iSCSI SAN and two NICs are used at the host, use Static discovery to control the host access to desired targets when using a 1GbE iSCSI Module option.
- To simplify configuration of the 1GbE iSCSI Module and meet adequate high availability, use Dynamic discovery at the ESX host when only two NICs are accessing the iSCSI SAN and up to two GbE ports are used per controller.

The 10GbE iSCSI Module option on the other hand provides two 10GbE iSCSI ports for LUN access. For this it is safe to use Dynamic discovery with a 10GbE iSCSI Module in all configurations as long as up to 2 NIC ports are used on the host side. Table 3 below provides guidance on selecting the right number of paths at the host and the array in a 10GbE iSCSI Module configuration with two distinct SANs as shown in Figure 11.

Table 3. HP EVA 10GbE iSCSI Module High Availability configuration decision chart

	Used 10GbE Port per controller	**1**		**2**	
	Used iSCSI targets used per LUN per 10GbE port	1	2	1	2
Used NIC ports per host	1	☒	☒	☒	☒
	2	✓	✓	✓	✓
	3	✓	✓	✓	x
	4	✓	✓	✓	x

☒ - Does not meet adequate high availability

✓ - Meet adequate path count and high availability

x - Exceeds the ESX supported Max path per LUN – Does not meet adequate or symmetrical high availability

All the same facts listed above for the 1GbE iSCSI Module apply to the 10GbE iSCSI Module. As in Table 2 above, the highlighted cell in Table 3 is the intersection of configuration options that provides the best balance of high availability and GbE port bandwidth for the 10GbE iSCSI Module configuration.

Best practices for 10GbE iSCSI Module configuration

- When all 10GbE ports are connected to the iSCSI SAN and two NICs are used at the host, use Dynamic discovery to simply configuration and meet adequate high availability with a 10GbE iSCSI Module option.

Performance consideration

When connecting ESX via iSCSI to an HP EVA, note that there is no performance gain from using different iSCSI LUN groups. As previously discussed the groups primarily provide additional LUN count access. For simplicity, it may be worthwhile separating iSCSI LUNs into different groups based on application or departmental usage for easier tracking of resources.

Because of the ESX limitation of eight paths per LUN for iSCSI, administrators need to carefully tradeoff their need for performance over availability. In a 1GbE iSCSI Module configuration, utilizing as many 1GbE Ethernet ports at the array side as possible will likely provide the highest performance boost but reduce the high availability of the system. In a 10GbE iSCSI Module configuration, an administrator can achieve higher availability by adding additional host connections to the iSCSI SAN with a smaller number of 10GbE ports while maintaining adequate performance.

Best practices for iSCSI performance and availability

- Using two iSCSI GbE ports at the array and two NICs on the ESX host provides the most flexible option to achieve the right balance of ease of configuration, high availability and performance for the 10GbE iSCSI Module configuration options.
- For the 1GbE iSCSI Module configuration option, using static discovery to limit the number of iSCSI targets used per GbE port to 1 can help achieve increased high availability and performance at the cost of ease of configuration.

Note

The HP EVA MPX200 configuration will work exactly as described above for 1GbE iSCSI Module and 10GbE iSCSI Module options. However, the MPX200 is capable of exporting LUNs for up to four different arrays at the same time.

The HP EVA does not currently support exporting a LUN to a host via iSCSI while simultaneously exporting this same LUN via Fibre Channel.

Implementing multi-pathing in vSphere 4.x and 5.x

A key task when configuring vSphere 4.x and 5.x is to set up multi-pathing so as to optimize the connectivity and operation of the EVA array. In addition, advanced tuning of HBAs, virtual SCSI adapters and ESX advanced parameters can help you increase storage performance.

vSphere 4 introduced the concept of path selection plug-ins (PSPs), which are essentially I/O multi-pathing options. These plug-ins are described in more detail in Configuring multi-pathing.

Table 4 outlines multi-pathing options in vSphere 4.x and 5.

Table 4. Multi-pathing options

I/O path policy	PSP	vSphere 4	vSphere 4.1	vSphere 5
MRU	VMW_PSP_MRU	Yes	Yes	Yes
Round Robin	VMW_PSP_RR	Yes	Yes	Yes
Fixed	VMW_PSP_FIXED	Yes	Yes	Yes
Fixed_AP (Array Preference)	VMW_PSP_FIXED_AP	No	Yes	(Fixed = Fixed_AP)[6]

[6] On vSphere 5.x, the FIXED I/O path policy when attached to an ALUA capable array operates like the FIXED_AP policy in vSphere 4.1.

The I/O path policies supported since vSphere 4.x are as follows:

- MRU
  - ALUA-aware
  - Gives preference to the optimal path to the Vdisk
  - If all optimal paths are unavailable, MRU uses a non-optimal path
  - When an optimal path becomes available, MRU fails over to this path
  - Although vSphere servers may use a different port through the optimal controller to access the Vdisk, only a single controller port is used for Vdisk access per vSphere server

- Round robin
  - ALUA-aware
  - Queues I/O to Vdisks on all ports of the owning controllers in a round robin fashion, providing instant bandwidth improvement over MRU
  - Continues queuing I/O in a round robin fashion to optimal controller ports until none are available, at which time it fails over to non-optimal paths
  - When an optimal path becomes available, round robin fails over to it
  - Can be configured to round robin I/O for a Vdisk to all controller ports by ignoring the optimal path preference

Note
Using round robin policy and ignoring the optimal path preference may be beneficial when you need to increase controller port bandwidth to accommodate a write-intensive workload.

- Fixed
  - Not ALUA-aware
  - Subject to the same intricate configuration considerations as ESX 3.5 or earlier
  - May result in a configuration where the non-optimal I/O path to a logical unit is used for I/O
  - Not recommended for use with vSphere 4.x and an EVA array

- Fixed_AP

  Introduced in vSphere 4.1, Fixed_AP I/O path policy extends the functionality of Fixed I/O path policy to active-passive and ALUA-compliant arrays. Fixed_AP can also identify the preferred controller for a Vdisk.

  Despite being ALUA-aware, the primary path selection attribute for Fixed_AP for ALUA capable arrays is the preferred controller for a Vdisk and not just its access state.

  To summarize, the key capabilities of Fixed_AP include:
  - ALUA-aware
  - Gives preference to the optimal path to a Vdisk
  - Changes the access state of a Vdisk but not its PREF setting
  - If all optimal paths are unavailable, Fixed_AP uses a non-optimal path and makes it optimal
  - If all non-optimal paths are unavailable, Fixed_AP uses an optimal path
  - If the path being used is non-optimal but preferred, Fixed_AP attempts to make this path optimal

- Although vSphere servers may use a different port through the optimal controller to the Vdisk, only a single controller port is used for Vdisk access per vSphere server
  - Starting with vSphere 5, the functionality of the FIXED_AP I/O path policy has been merged with the legacy Fixed I/O path policy. So when connected to an ALUA -capable storage array, the Fixed I/O path policy in ESXi 5.x will behave like Fixed_AP did in ESX 4.1 and revert to legacy Fixed I/O path policy when connected to non-ALUA storage arrays.

Implementing multi-pathing

Since vSphere 4.x and 5.x are ALUA-compliant, their implementation of multi-pathing is less complex and delivers higher levels of reliability than ESX 3.5 or earlier.

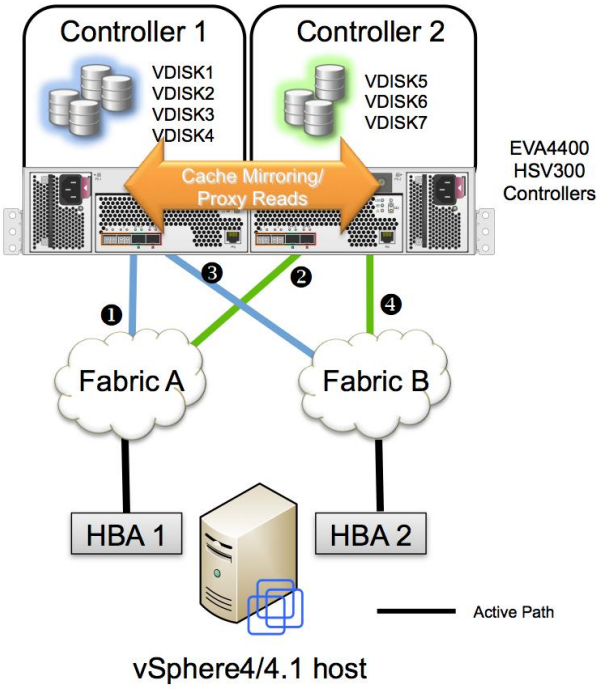
Setting up multi-pathing only requires the following steps:

- Configure the Vdisk
- Select the controller access policy at the EVA
- Power on/reboot all vSphere 4.x/5.x servers or perform a rescan

Following the boot or rescan, vSphere 4.x detects the optimal access paths and, if MRU, round robin I/O or Fixed_AP path policy has been selected, gives optimal paths the priority for issuing I/Os.

Figure 14 shows a typical multi-pathing implementation using vSphere 4.x/5.x.

Figure 14. EVA connectivity with vSphere 4.x/5.x

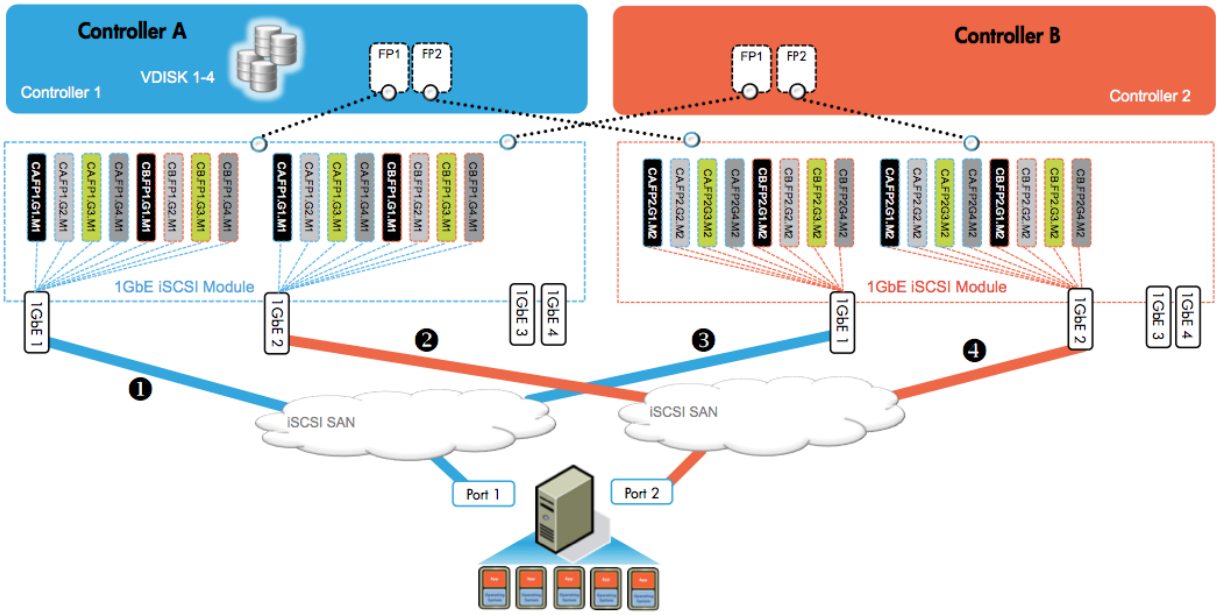


All I/Os to Vdisks 1 – 4 are routed through one or more ports on Controller 1 and through Paths ❶ and/or ❸, regardless of the HBA that originates the I/O. Similarly, all I/Os to Vdisks 5 – 7 are routed to Controller 2 through Paths ❷ and/❹, regardless of the originating HBA.

The vSphere 4.x/5.x implementation yields much higher system resource utilization and throughput and, most importantly, delivers a balanced system out of the box, with no intricate configuration required.

When configuring HP EVA with iSCSI Modules, it must be noted that I/O will be routed slightly differently than with a Fibre Channel configuration. With an HP EVA iSCSI configuration, I/O is still routed to the same controller that is the optimal path to Vdisks 1-4 however the data flows through the iSCSI Module in the alternate controller before being routed back to the LUNs owning controller. Figure 15 below illustrates this.

Figure 15. EVA iSCSI connectivity with vSphere 4.x/5.x



Paths 1 and 3 are the optimal paths to Vdisks 1-4. However path 3 has a connection through the iSCSI Module that is embedded in controller B. Internal to the iSCSI Module, I/O coming through path 3 is routed to controller A.

Note
From the perspective of Vdisk access, you can easily balance vSphere 4.x with an EVA array. However, it may be desirable to capture performance metrics at the EVA and assess data access to/from each controller to ensure that the workload is truly balanced between the two controllers.

Using Fixed_AP I/O path policy

Because it is ALUA-aware, Fixed_AP I/O path policy can extend the functionality of Fixed I/O path policy to active-passive and ALUA-aware arrays. In addition, Fixed_AP can identify the preferred controller for a Vdisk.

Despite being ALUA-aware, Fixed_AP's primary path selection attribute is the preferred controller setting of a Vdisk and not just its access state.

Note that the preferred controller for accessing a Vdisk in an ALUA-capable array is defined in SCSI by the PREF bit, which is found in byte 0, bit 7 of the target port descriptor format[7]. If the PREF bit is set to one, this indicates that the Vdisk is preferred to the controller the target port group request was sent to; if it is set to zero, this indicates the Vdisk is not preferred to the controller the target port group request was sent to. Thus, the controller preference in an EVA array is equivalent to setting a Vdisk access path in Command View EVA to Path A/B-Failover/failback, as described in Vdisk provisioning.

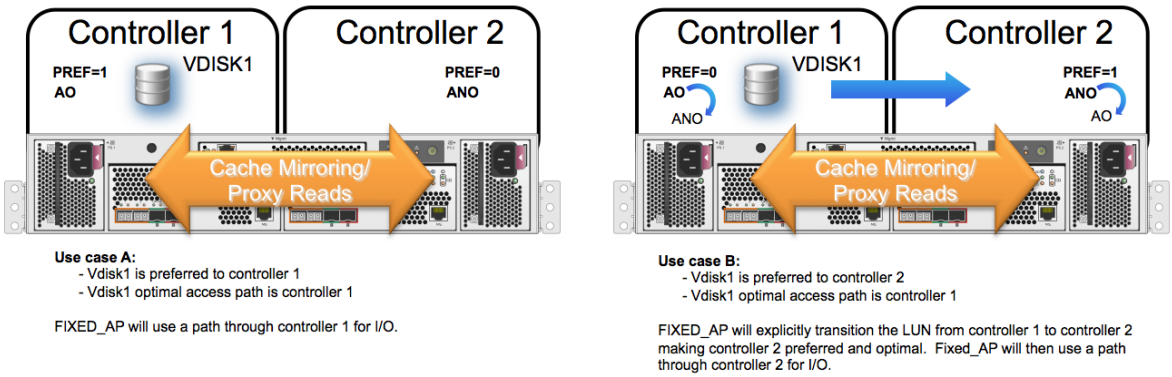
Primary use case

The primary use case for Fixed_AP is based on its ability to automatically return to a balanced, pre-configured environment after Vdisk access has become unbalanced following an event such as a controller failure or restore. After a controller failure, all Vdisks would migrate to the remaining controller and become optimized for that controller; however the logical unit's initial controller preference does not change and can point to the restored controller. Fixed_AP causes all Vdisks with a preference for the restored controller to migrate back to that controller.

Consider the uses cases shown in Figure 16.

[7] For more information, refer to SCSI Primary Commands (SPC) standard.

Figure 16. Fixed_AP use cases.



Fixed_AP can cause explicit Vdisk transitions to occur and, in a poorly configured environment, may lead to Vdisk thrashing.

Since transitioning Vdisks under heavy loads can have a significant impact on I/O performance, the use of Fixed_AP is not recommended for normal production I/O with EVA arrays.

Note
Fixed_AP can only be used to explicitly transition Vdisks on storage systems that support explicit transitions, such as EVA arrays.

Fixed_AP can be leveraged to quickly rebalance preferred access to Vdisks after the configuration has become unbalanced by controller failure/restoration or implicit transitions triggered by the storage and others.

Summary
In vSphere 4.x/5.x, ALUA compliance and support for round robin I/O path policy have eliminated the intricate configuration required to implement multi-pathing in ESX 3.5 or earlier. These new features also help provide much better balance than you could achieve with MRU; furthermore, round robin policy allows I/Os to be queued to multiple controller ports on the EVA, helping create an instant performance boost.

Best practices for I/O path policy selection

- Round robin I/O path policy is the recommended setting for EVA asymmetric active-active arrays. MRU is also suitable if round robin is undesirable in a particular environment.
- Avoid using legacy Fixed I/O path policy with vSphere 4.x and EVA arrays.
- In general, avoid using Fixed_AP I/O path policy with vSphere 4.x and EVA. However, this policy can be leveraged to quickly rebalance Vdisks between controllers – for example, after a controller has been restored following a failure. This use case can be employed with a single vSphere host when the array is not under heavy load. Once the balanced state has been restored, you should end the use of Fixed_AP and replace it with a recommended path policy.
- Avoid using Fixed I/O path policy with vSphere 5.x and EVA. However since Fixed I/O path policy in ESX 5.x is the same as Fixed_AP in ESX 4.1, the same use case considerations discussed above with ESX 4.1 also apply to Fixed I/O path policy in ESX 5.x.

Configuring multi-pathing

The multi-pathing framework for vSphere 4.x/5.x includes the following core components:

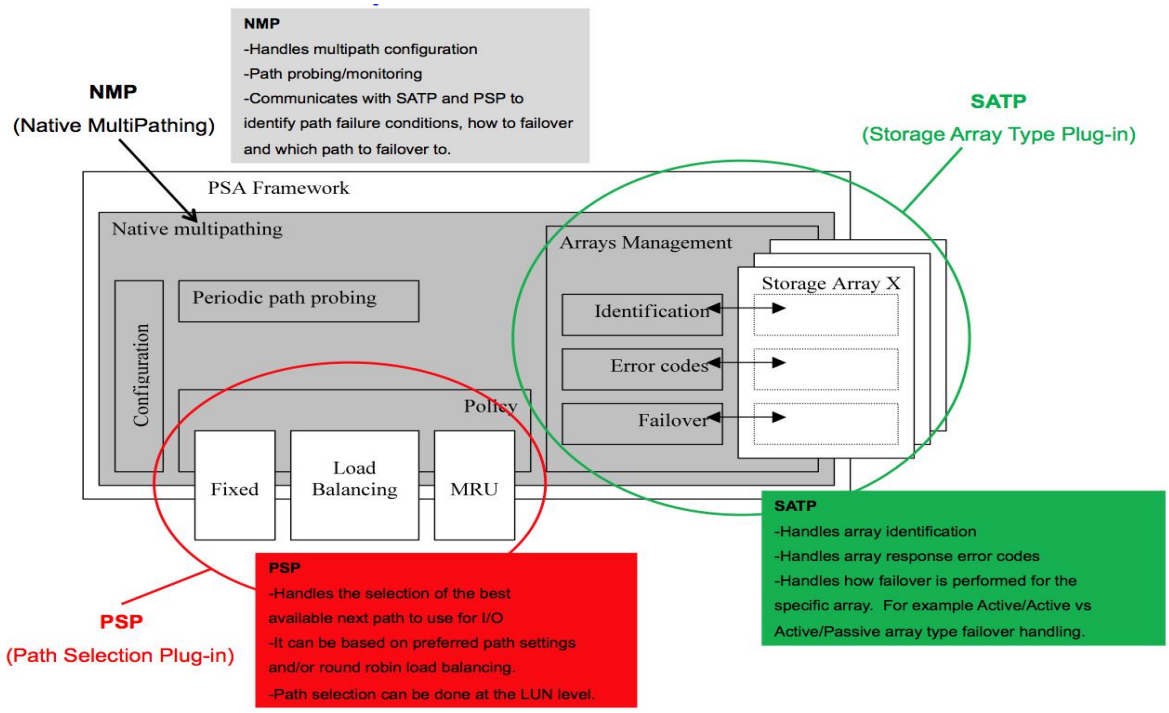
- Native Multi-pathing Plug-in (NMP)
  Also known as the NMM (Native Multi-pathing management extension module (MEM))
- Storage Array Type Plug-in (SATP)
  Also known as the SATM (Storage Array Type MEM); used in conjunction with the NMP
- Path Selection Plug-in (PSP)
  Also known as the PSM (Path Selection MEM); used in conjunction with a specific SATP

- **Multi-Pathing Plug-in** (MPP)

  Third-party implementation (which is outside the scope of this document) that takes the place of the NMP/SATP/PSP combination

Figure 17 outlines key components of the multi-pathing stack.

Figure 17. vSphere 4.x and 5.x multi-pathing stack



The key features of the multi-pathing plug-ins are as follows:

- **SATP**

  The SATP is an array-specific plug-in that handles specific operations such as device discovery, the management of array-specific error codes, and failover.

  For example, while storage arrays use a set of standard SCSI return codes to warn device drivers of various failure modes, they also make use of vendor-specific codes to handle proprietary functions and/or behavior. The SATP takes the appropriate action when these vendor–specific return codes are received.

- **PSP**

  The PSP selects an appropriate path to be used to queue I/O requests. PSP utilizes the following I/O path selection policies:
  - Fixed
  - Fixed_AP[8]
  - MRU
  - Round robin

  PSP settings are applied on a per-Vdisk basis; thus, within the same array, it is possible to have some Vdisks using MRU while others are using the round robin policy.

- **NMP**

  The NMP ties together the functionality delivered by the SATP and PSP by handling many non-array specific activities, including:
  - Periodical path probing and monitoring
  - Building the multi-pathing configuration

[8] Only supported in vSphere 4.1

When a path failure occurs, the NMP communicates with the SATP and PSP and then takes the appropriate action. For example, the NMP would update its list of available paths and communicate with the PSP to determine how I/O should be re-routed based on the specified path selection policy.

Displaying the SATP list

Although there is no way to monitor the NPM's list of available paths through vCenter, you are able to display a list of available SATPs and their respective default PSPs (as shown in Table 5).

Use the following CLI command on the ESX console:

```
esxcli nmp satp list/grep EVA
```

Note
Currently, you cannot reconfigure the SATP rules table through vCenter.

Table 5. vSphere 4.x and 5.x SATP rules table, with entries that are relevant to EVA arrays denoted by an asterisk

SATP	Default PSP	Description
VMW_SATP_ALUA_CX	VMW_PSP_FIXED	Supports EMC CLARiiON CX arrays that use ALUA
VMW_SATP_SVC	VMW_PSP_FIXED	Supports IBM SAN Volume Controller (SVC) appliances
VMW_SATP_MSA	VMW_PSP_MRU	Supports HP Modular Smart Array (MSA) arrays
VMW_SATP_EQL	VMW_PSP_FIXED	Supports EqualLogic arrays
VMW_SATP_INV	VMW_PSP_FIXED	Supports the EMC Invista application
VMW_SATP_SYMM	VMW_PSP_FIXED	Supports EMC Symmetrix arrays
VMW_SATP_LSI	VMW_PSP_MRU	Supports LSI and other arrays compatible with the SIS 6.10 in non-AVT mode
VMW_SATP_EVA*	VMW_PSP_FIXED	Supports HP EVA arrays
VMW_SATP_DEFAULT_AP*	VMW_PSP_MRU	Supports non-specific active/passive arrays
VMW_SATP_CX	VMW_PSP_MRU	Supports EMC CLARiiON CX arrays that do not use ALUA protocol
VMW_SATP_ALUA*	VMW_PSP_MRU	Supports non-specific arrays that use the ALUA protocol
VMW_SATP_DEFAULT_AA	VMW_PSP_FIXED	Supports non-specific active/active arrays
VMW_SATP_LOCAL	VMW_PSP_FIXED	Supports direct-attached devices

IMPORTANT
SATPs are global, whereas a PSP can either be global or set on a per-Vdisk basis. Thus, a particular array can only use a specific SATP; however, Vdisks on this array may be using multiple PSPs – for example, one Vdisk can be set to round robin I/O path policy, while another Vdisk on the same array is set to MRU.

As indicated in Table 5, the following SATPs are relevant to EVA arrays:

- VMW_SATP_DEFAULT_AP – This SATP is used by active/passive EVA arrays. However, since such arrays are not supported in vSphere 4.x and 5.x, you should not use this plug-in to enable a vSphere 4.x and 5.x connection to an active/passive EVA for production purposes.
- VMW_SATP_EVA – This SATP is intended for active/active EVA arrays that have the Target Port Group Support (ALUA compliance) option turned off; however, all such arrays have TPGS turned on by default. Since this option is not user-configurable, no current EVA arrays use VMW_SATP_EVA.
- VMW_SATP_ALUA – This SATP is intended for any ALUA-compliant array; thus, it is used with active-active EVA arrays.

Connecting to an active-active EVA array in vSphere 4.0, 4.1 and 5.x

When connecting a vSphere 4.x or 5.x host to an active-active EVA array, you should use the VMW_SATP_ALUA SATP as suggested above. This SATP is, by default, associated with VMW_PSP_MRU, a PSP that uses MRU I/O path policy.

There are two steps for connecting a vSphere 4.x and 5.x server to the EVA array:

- Change the default PSP for the VMW_SATP_ALUA from VMW_PSP_MRU to VMW_PSP_RR
- Update an advanced configuration parameter for the VMW_PSP_RR PSP

Changing the PSP

First, you must change the VMW_SATP_ALUA default PSP/PSM from MRU to round robin.

PSPs in vSphere 4.x and 5.x are set at the Vdisk level and are based on an SATP. Since all active-active EVA arrays use the VMW_SATP_ALUA plug-in, configuring the VMW_SATP_ALUA default PSP to VMW_PSP_RR causes every new Vdisk from the EVA array to be configured automatically to use round robin path policy.

Make the change via the ESX CLI using the following command using ESX 4.x:

```
esxcli nmp satp setdefaultpsp -s VMW_SATP_ALUA -P VMW_PSP_RR
```

With ESXi 5.x, the esxcli command name space has changed. To perform the same operation on ESXi 5.x as shown above for ESX 4.x, you must issue the following command:

```
esxcli storage nmp satp set –s VMW_SATP_ALUA –P VMW_PSP_RR
```

Note
When this setting is applied on the fly, it only affects new Vdisks added to the vSphere host. In order for the change to affect all Vdisks (including pre-existing logical units), a reboot of the host is recommended.
Alternatively, you could unclaim and reclaim all devices managed by the NMP.

Best practice for changing the default PSP option in vSphere 4.x/5.x

- For an EVA array environment, change the default PSP option for the VMW_SATP_ALUA SATP to VMW_PSP_RR.

On running systems where a reboot or unclaiming/reclaiming the devices is not feasible, the PSP setting can be changed on the fly for each Vdisk individually running the command below:

For ESX 4.x:

```
esxcli nmp device setpolicy –P VMW_PSP_RR -d naa.xxxxxxxxx
```

For ESXi 5.x:

```
esxcli storage nmp device set –P VMW_PSP_RR -d naa.xxxxxxxxx
```

Alternatively, the following scripts can be run to apply this setting to all your EVA Vdisks at once. Before running the scripts below, ensure that you only have EVA Vdisks connected to ESX or that setting Round Robin path policy to all SAN devices is your intended goal:

For ESX 4.x:

```
for i in 'esxcli nmp device list | grep ^naa.6001' ; do esxcli nmp device
setpolicy –P VMW_PSP_RR -d $i; done
```

For ESXi 5.x:

```
for i in 'esxcli storage nmp device list | grep ^naa.6001' ; do esxcli storage nmp
device set –P VMW_PSP_RR -d $i; done
```

Updating the new PSP

To optimize EVA array performance, HP recommends changing the default round robin load balancing IOPS value to 1. This update must be performed for every Vdisk using the following command on ESX 4.x:

```
esxcli nmp roundrobin setconfig -t iops -I 1 -d naa.xxxxxxxxx
```

or the following command on ESXi 5.x

```
esxcli storage nmp psp roundrobin deviceconfig set –t iops –I 1 –d naa.xxxxxxxxx
```

In an environment where you only have EVA Vdisks connected to vSphere 4.x/5.x hosts you can use the following script to automatically set I/O path policy for each Vdisk to round robin:

For ESX 4.x:

```
for i in 'esxcli nmp device list | grep ^naa.6001' ; do esxcli nmp roundrobin
setconfig -t iops –I 1 –d $i; done
```

For ESXi 5.x:

```
for i in 'esxcli storage nmp device list | grep ^naa.6001' ; do esxcli storage nmp
psp roundrobin deviceconfig set –t iops –I 1 –d $i; done
```

Note that all the Vdisks must previously be configured to use the Round Robin Path Policy before running the scripts above. Refer to Changing the PSP section above for instruction to change the PSP setting for the Vdisks.

For environments with multiple array models, merely change **grep naa.6001** so that it matches the pattern to devices on the desired arrays only.

Caveats for connecting to vSphere 4.1 and 5.x

vSphere 4.0 does not provide a method for globally configuring the path selection plug-in (PSP or PSM) based on a particular array model. However, starting with vSphere 4.1, VMware introduced more granular SATP and PSP configuration options that allow array specific SATP and PSP configurations.

As in vSphere 4, each SATP has a default PSP. However, vSphere 4.1/5.x also gives you the ability to configure a particular PSP based on the storage array model. This is a significant improvement, allowing multiple arrays to use the same SATP but, by default, utilize different PSPs, which provides a tremendous savings in configuration time.

As in vSphere 4, the default PSP for VMW_SATP_ALUA in vSphere 4.1/5.x is VMW_PSP_MRU. Configuring the recommended PSP, VMW_PSP_RR, can be achieved in two different ways, depending on the deployment type:

- **Same I/O path policy settings**

  If the vSphere 4.1/5.x cluster is connected to one or more ALUA-capable arrays using the same I/O path policy, you can change the default PSP for the VMW_SATP_ALUA to VMW_PSP_RR with the same steps described in the section Connecting to an active-active EVA array in vSphere 4.0, 4.1 and 5.x shown above.

- **Different I/O path policy settings**

  If the vSphere 4.1/5.x cluster is connected to two or more ALUA capable arrays that each requires different I/O path policy settings, you can leverage the new SATP and PSP configuration options in vSphere 4.1/5.x via the following command line:

  For ESX 4.1:

  ```
  esxcli nmp satp addrule –s VMW_SATP_ALUA –P VMW_PSP_RR –o iops –O iops=1 –c
  tpgs_on –V HP –M HSV300 –e "My custom EVA4400 rule"
  ```

  For ESXi 5.x:

  ```
  esxcli storage nmp satp rule add –s VMW_SATP_ALUA –P VMW_PSP_RR –o iops –O iops=1
  –c tpgs_on –V HP –M HSV300 –e "My custom EVA4400 rule"
  ```

  This single command line achieves the following:

  – Create a new rule in the SATP rule table for the array specified with **–vendor** and **–model**

  – Set the default SATP for this array to **VMW_SATP_ALUA**

  – Set the default PSP to **VMW_PSP_RR**

  – Set the round robin option to **IOPS=1**

  Repeat this command line for each ALUA-compliant array model to be shared by your vSphere 4.1/5.x cluster.

  With this single command line, you can achieve the same results as the two-stage configuration process required for vSphere 4. Thus, regardless of the deployment type in vSphere 4.1/5.x, running this single **addrule** command is far more efficient.

Use the following command to verify that the new rule has been successfully added:

```
esxcli nmp satp listrules
```

Deleting a manually-added rule
To delete a manually-added rule, use the **esxcli nmp satp deleterule** command; specify the same options used to create the rule. For example:

```
esxcli nmp satp deleterule --satp="VMW_SATP_ALUA" --psp="VMW_PSP_RR" --psp-
option="iops=1" --claim-option="tpgs_on" --vendor="HP" --model="HSV210" --
description="My custom HSV210 rule"
```

Caveats for changing the rules table
When making changes to the SATP rules table, consider the following:

- On-the-fly rule changes only apply to Vdisks added to the particular array after the rule was changed. Existing Vdisks retain their original settings until a vSphere server reboot occurs or a path reclaim is manually triggered.
- The array vendor and models strings used in the **addrule** command line must exactly match the strings returned by the particular array. Thus, if the new rule does not claim your devices – even after a server reboot – then verify that the vendor and model strings are correct.
- As you add rules to the SATP rules, tracking them can become cumbersome; thus, it is important to always create a rule with a very descriptive, consistent description field. This facilitates the retrieval of user-added rules using a simple filter.

Best practice for changing the default PSP in vSphere 4.1/5.x

- Create a new SATP rule for each array model.

Best practice for configuring round robin parameters in vSphere 4.x/5.x

- Configure IOPS=1 for round robin I/O path policy.

Caveats for multi-pathing in vSphere 4.x/5.x

This section outlines caveats for multi-pathing in vSphere 4.x associated with the following:

- Deploying a multi-vendor SAN
- Using Microsoft clustering
- Toggling I/O path policy options
- Using data replication (DR) groups
- Using third-party multi-pathing plug-ins

Deploying a multi-vendor SAN
In an environment where ALUA-capable arrays from multiple vendors are connected to the same vSphere 4.x cluster, exercise caution when setting the default PSP for VMW_SATP_ALUA, especially if the arrays have different recommendations for the default PSP option.

Setting the appropriate PSP differs depending on the deployment type.

vSphere 4 deployment
If the total number of EVA Vdisks is smaller than the total number of logical units from third-party arrays, then set the default PSP option to VMW_PSP_RR, assuming that the third-party storage vendor(s) also recommend(s) this setting. Otherwise, use the recommended default for the third-party arrays and manually configure EVA Vdisks. Thus, to minimize configuration time, the bulk of the task is automatically performed by default using vSphere 4.x, leaving you to manually run a simple script to set the desired PSP – VMW_PSP_RR – for EVA Vdisks.

The above recommendation only applies to the following use case:

- The EVA and third-party arrays are in the same vSphere 4 SAN
- Arrays from two or more vendors are ALUA-compliant
- There are different default recommendations for PSPs

vSphere 4.1/5.x deployment

- If the multi-vendor SAN is being shared by a vSphere 4.1 cluster, then create a new SATP rule entry for each array, setting the configuration parameters as recommended by the particular vendor.

Best practice for configuring PSP in a multi-vendor SAN

- vSphere 4: When using vSphere 4 in a multi-vendor, ALUA-compliant SAN environment, configure the default PSP for the VMW_SATP_ALUA SATP to the recommended setting for the predominant array type or to the recommended setting for the array type with the most Vdisks provisioned for vSphere access.
- vSphere 4.1/5.x: Create a SATP rule entry for each storage array with the desired attributes.

Using Microsoft clustering

At the time of writing, VMware does not support round robin I/O path policy in conjunction with VMs clustered via Microsoft Cluster Server (MSCS). Thus, HP recommends setting all VMware Raw Device Mapping (RDM) Vdisks used by an MSCS cluster to utilize MRU as the preferred I/O path policy.

Since most MSCS cluster deployments utilize several logical units, you should configure the path policy for these Vdisks manually.

Best practice for configuring MSCS cluster Vdisks

- MSCS cluster Vdisks should be configured to use the MRU I/O path policy. Since the recommended default setting for all EVA Vdisks is round robin, you must manually configure these Vdisks to MRU.

Toggling I/O path policy options

Once configured, avoid toggling the I/O path policy for Vdisks using round robin policy. When I/O path policy is changed from round robin to any other (fixed or MRU), either through vCenter or a CLI, all round robin advanced configuration settings are lost. The next time you set round robin policy for that device, you must manually reset these settings.

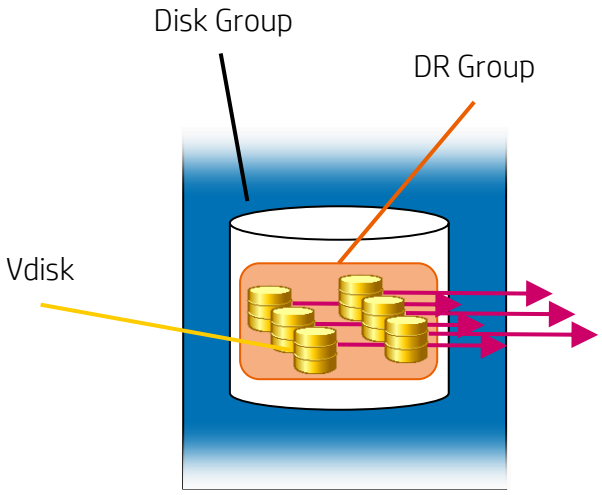
Using data replication groups

The HP Continuous Access EVA software allows data to be replicated between two or more EVA arrays, either synchronously or asynchronously.

Continuous Access EVA supports various interconnection technologies such as Fibre Channel or Fibre Channel over IP (FCIP).

A DR group is the largest replication object within Continuous Access EVA and is comprised of replicated Vdisks (copy sets), as shown in Figure 18.

Figure 18. Relationship between Vdisks and the DR group



Just like a Vdisk, a DR group is managed through one controller or the other; in turn, this controller must manage all Vdisks within the particular DR group. Thus, when a Vdisk is added to an existing DR group, its optimal controller is the one currently managing the DR group.

Since the inheritance of controllers can impact the overall balance of Vdisk access, you should ensure that DR groups are spread across both controllers.

Best practice for managing controllers for DR groups

- Ensure that DR groups are spread between the controllers so as to maintain an adequate balance.

Using third-party multi-pathing plug-ins (MPPs)

vSphere 4.x/5x allow third-party storage vendors to develop proprietary PSP, SATP, or MPP plug-ins (or MEMs).

These third-party MEMs are offered to customers at an incremental license cost and also require enterprise VMware licensing.

The EVA array was designed to provide optimal performance and functionality using native VMware multi-pathing plug-ins, saving you the extra work and expense associated with proprietary plug-ins. When used, configured and tuned appropriately, native plug-ins can significantly reduce configuration time and provide enhanced performance in most environments at zero incremental cost, while keeping the solution simplified.

Upgrading EVA microcode

An online upgrade of EVA microcode is supported with vSphere 4.x/5.x.

When performing such upgrades it is critical to follow the general EVA Online Firmware Upgrade (OLFU) guidelines defined in the OLFU best practices guide[9].

From a vSphere 4.x/5.x perspective, VMs using RDM Vdisks are more susceptible to issues resulting from an OLFU. It is important to ensure that the SCSI disk timeout for all VMs is set to a minimum of 60 seconds, or higher (60 – 90 seconds) in a larger environment.

Guidelines are provided for setting the SCSI disk timeout for Microsoft Windows and Linux VMs.

Setting the timeout for Windows VM

For a VM running Windows Server 2003[10] or earlier, change the value of the HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Disk/TimeoutValue registry setting to 3c (that is, 60 expressed in hexadecimal form).

A reboot is required for this change to take effect.

Setting the timeout for Linux VMs

Use one of the following commands to verify that the SCSI disk timeout has been set to a minimum of 60 seconds:

```
cat /sys/bus/scsi/devices/W:X:Y:Z/timeout
```

or

```
cat /sys/block/sdX/device/timeout
```

If required, set the value to 60 using one of the following commands:

```
echo 60 > /sys/bus/scsi/devices/W:X:Y:Z
```

or

```
echo 60 | cat /sys/block/sdX/device/timeout
```

where W:X:Y:Z or sdX is the desired device.

No reboot is required for these changes to take effect.

Overview of vSphere 4.x/5.x storage

vSphere 4.x/5.x support VMware Virtual Machine File System (VMFS), RDM, and Network File System (NFS) datastores, each of which can deliver benefits in a particular environment. This section provides information for the following topics:

- Using VMFS
- Using RDM
- Comparing supported features
- Implementing a naming convention
- Sizing the vSphere cluster
- Aligning partitions

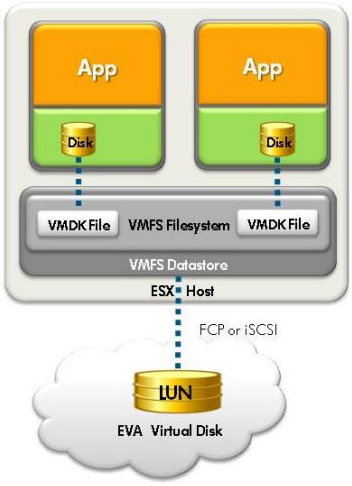
[9] See http://h20000.www2.hp.com/bc/docs/support/SupportManual/c01921588/c01921588.pdf
[10] In Windows Server 2008, the SCSI timeout defaults to 60 seconds.

Using VMFS

VMFS is a high-performance cluster file system designed to eliminate single points of failure, while balancing storage resources. This file system allows multiple vSphere 4.x hosts to concurrently access a single VMDK (Virtual Machine Disk Format), as shown in Figure 19.

VMFS supports Fibre Channel SAN, iSCSI SAN, and NAS storage arrays.

Figure 19. VMFS datastore



Best practices for deploying VMFS

- To avoid spanning VMFS volumes, configure one VMFS volume per logical unit.
- In general limit VMs or VMDKs to 15-20 per volume.
- Either place I/O-intensive VMs on their own SAN volumes or use RDMs, which can minimize disk contention.

  When placing I/O-intensive VMs on a datastore, start with no more than six to eight vSphere hosts per datastore. Monitor performance on the vSphere hosts to ensure there is sufficient bandwidth to meet your application requirements and that latency is acceptable.

Using RDM

As shown in Figure 20, RDM allows VMs to have direct access to Vdisks, providing support for applications such as MSCS clustering or third-party storage management.

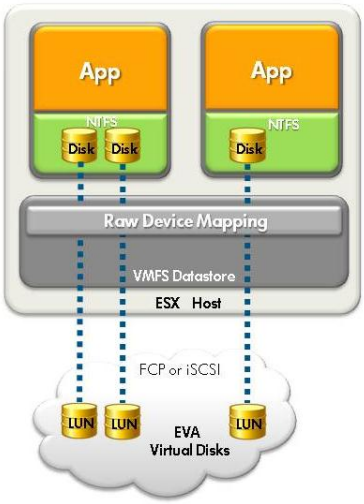
vSphere 4.x/5.x provides the following RDM modes, which support advanced VMware features like vMotion, High Availability (HA), and Distributed Resource Scheduler (DRS):

- **Virtual compatibility mode** (vRDM):
  - All I/O travels through the VMFS layer
  - RDMs can be part of a VMware snapshot
- **Physical compatibility mode** (pRDM):
  - All I/O passes directly through the underlying device
  - pRDM requires the guest to use the virtual LSI Logic SAS controller
  - pRDM is most commonly used when configuring MSCS clustering

There are some limitations when using RDM in conjunction with MSCS or VMware snapshots – for example, when configuring an MSCS cluster between a physical Windows Server and a Windows virtual machine, you should utilize pRDM because vRDM is not supported in that configuration. However, for a cluster between Windows Virtual Machine residing on the same host (aka cluster in a box) you can use both vRDM and pRDM for the cluster configuration. Note that when configuring Windows 2008 VMs in a cluster, you must use the LSISAS Virtual adapter for the shared pRDM.

For more information, refer to the VMware white paper, Setup for Failover Clustering and Microsoft Cluster Service.

Figure 20. RDM datastore



Comparing supported features

Table 6 compares features supported by VMFS and RDM datastores.

Table 6. Datastore supported features

Feature	VMFS	RDM
VMware Native Multi-pathing (NMP)	Yes	Yes
VMware vMotion	Yes	Yes
VMware Storage vMotion	Yes	Yes
VMware FT	Yes	Yes
VMware sDRS	Yes	No
VMware SIOC	Yes	No
MSCS	Yes[11]	Yes

Implementing a naming convention

When working with multiple datastores, VMDKs, and RDMs in conjunction with large clusters or SANs, management can become complex. Unless you use a suitable naming convention, it may be very difficult to locate a particular datastore for management or troubleshooting purposes.

For example, array model names should not be included in the naming convention – if you move a Vdisk from one EVA model (such as the EVA8100) to another (such as the EVA4400), the naming convention would break down. You should not include the Vdisk number in a datastore or RDM naming convention because the number for a Vdisk in Datacenter A may not be maintained when the Vdisk is replicated and presented to a host in Datacenter B. Similarly, you should not include Vdisk size in a naming convention because it is probable that this size will change.

To avoid confusion, you need detailed documentation on each array, datastore, Worldwide Name (WWN), and host name. In addition, avoid using the following in the name for a datastore or RDM:

- EVA model or controller name
- EVA WWN (or any part thereof)
- Vdisk number

[11] Only supported with vRDM using VMs in a cluster within a box configuration.

- Vdisk size
- Vdisk WWN
- Name of the vSphere server name from which the datastore was created

Creating the convention
HP recommends naming VMFS datastores and RDMs in vCenter with the same name used when creating the Vdisk in Command View EVA or when using SSSU scripting tools. While this approach is beneficial, it may require some coordination with the storage administrator.

Note
When creating Vdisks in Command View or SSSU, Vdisk names are unique to the particular array.

In a vSphere cluster, ESX prevents you from creating datastores with the same name, regardless of the array they are created on. This capability yields an environment where datastore names are unique across the entire infrastructure, while RDM names are unique to the particular vSphere host – or, depending on your choice of names, to the entire infrastructure.

Using descriptive naming
When creating Vdisks in Command View EVA, HP recommends using descriptive naming for the datastore. For example, a suitable naming convention might be as follows:

<Location>_<Location Attribute>_<Department>_<Vdisk Type>_<Usage>_#

Table 7 outlines the various components of this naming convention.

Table 7. Sample naming convention

Component	Description	Example
<Location>	Geographical location of the team, group or division for which the storage is being allocated	The location could be a city, state, country, or a combination thereof.
<Location Attribute>	Specific attribute of the particular location, such as an office building floor or the scope of the department	• Building R5 • Third floor
<Department>	Job function of the team, group or division for which the storage is being allocated	• IT • Sales • Finance
<Vdisk Type>	Type of Vdisk	• RDM • VMFS (datastore)
<Usage>	How the Vdisk will be used	• VM boot • Exchange logs • Exchange data • Oracle_archLogs • Oracle_data
<#>	Provides enumeration in the event that multiple disks with the same name should be needed	• 1 • 0002

The following are examples of this sample naming convention:

- Houston_Enterprise_Sales_VMFS_VMBoot_1
- Houston_Enterprise_Sales_VMFS_VMBoot_2
- LA_Corporate_IT_RDM_ExchLogs_1
- LA_Corporate_IT_RDM_ExchLogs_2

The advantages of this naming convention are as follows:

- Simple
- Synchronizes EVA and vSphere naming
- Independent of any unique array or Vdisk properties
- Does not break due to replication or Storage vMotion
- Uniqueness automatically ensured by EVA

However, there are challenges associated with the naming convention, including the following:

- Inability to quickly match the array Vdisk number only using the Vdisk name
- Cross-reference matrix required for information such as Vdisk ID and array

Both of the above challenges can be addressed by deploying the HP Insight Control Storage Module for vCenter plug-in.

Best practice for naming datastores

- When naming a datastore, utilize the same name used in Command View when the Vdisk was created.
- Use simple, consistent names for your datastores – in the future, you may need to add vSphere hosts to the cluster.

Sizing the vSphere cluster

Knowing how many vSphere hosts can be supported per Vdisk will enhance your ability to manage the cluster[12].

Monitor resource utilization to recognize VMs that are I/O-intensive. Often, high I/O requirements will lower your overall host count.

Best practice for using Vdisk IDs

- To make it easier to manage the cluster and SAN, use the same Vdisk ID for all vSphere hosts in the same cluster.

Aligning partitions

In a vSphere environment, partitions on VM data disks must be aligned with disks on the EVA array; however, the EVA array has no alignment requirements. See the VMware vStorage API for Array Integration section below for VAAI specific alignment considerations and best practices.

Aligning Vdisk and VMFS partitions

The array is a virtualized storage system that is managed by Command View. When you use Command View to create a Vdisk to present to a supported host, the host mode you specify ensures that the Vdisk is appropriately aligned for the file system you plan to mount.

When you present a Vdisk to a host, you can use your favorite secure shell (ssh) client to login to your vSphere host and view the configuration of the presented Vdisk; you can then format the Vdisk with a VMFS filesystem. Using the vSphere client to create the VMFS filesystem automatically aligns the VMFS filesystem partition with the underlying Vdisk.

Aligning VM operating system partitions

IMPORTANT
If you are using Microsoft Windows Vista®, Windows 7, or Windows Server 2008, you do not need to align partitions. These operating systems correctly align the partition at the time you create the VMDK.

At some stage of your deployment, you may need to align the partitions for certain VM operating systems with the VMDK to avoid performance degradation in your VM or application.

For EVA arrays, HP recommends creating all your VMDK files via the vSphere client and verifying that each Vdisk is appropriately aligned; other storage systems may be different.

If you are using Storage vMotion to move your datastores, consult the appropriate storage vendor to ensure your VMFS partitions or VMDK files are correctly aligned to disk.

[12]The maximum number of Vdisks that can be connected to a vSphere cluster is 256.

VMware has carried out testing to compare I/O performance with aligned and non-aligned file systems and, as a result, suggests working with your vendor to establish the appropriate starting boundary block size.

Best practices for aligning the file system

- No alignment is required with Windows Vista, Windows 7, or Windows Server 2008.
- Use the vSphere client when creating your datastore, which correctly aligns the file system.
- Verify that VMDKs used by the guest operating system are correctly aligned.

Enhancing storage performance

A range of vSphere 4.x/5.x features and tuning options provide opportunities for enhancing storage performance. This section provides information on the following topics:

- Optimizing queue depth
- Using adaptive queuing
- Using the paravirtualized virtual SCSI driver
- Monitoring EVA performance in order to balance throughput
- Optimizing I/O size

Optimizing queue depth

Queue depth is a function of how quickly processes are loaded into the queue and how fast the queue is emptied between the HBA and a Vdisk.

Tuning the Vdisk queue depth is often regarded as a requirement in a vSphere environment that uses SAN storage. Indeed, in some deployments, tuning the queue may help to enhance performance; however, injudicious tuning can result in increased latency. HP suggests that, if your storage system is properly configured and balanced, default queue depth values may be ideal. The best approach is to analyze the entire environment, not just a single vSphere host.

Adjusting the queue depth requires you to determine how many commands the HBA can accept and process for a given logical unit. Thus, as a best practice when adjusting the HBA queue depth, you should adjust the vSphere Disk.Sched.NumReqOutstanding setting for each vSphere host. The simplest way is to use the vSphere client and login to vCenter to make the necessary adjustments.

Best practice for adjusting the queue depth

- When increasing the HBA queue depth, also increase the vSphere Disk.Sched.NumReqOutstanding setting.

Using adaptive queuing

ESX 3.5 and vSphere 4.x/5.x provide an adaptive queue depth algorithm that can dynamically adjust the logical unit (LU) queue depth in the VMkernel I/O stack when congestion is reported by the storage system in the form of QUEUE FULL or DEVICE BUSY status.

Dynamically adjusting the LU queue depth allows the VMkernel to throttle back I/O requests sent to the particular LU, thus reducing I/O congestion in the storage system.

Adaptive queuing is controlled by the following advanced parameters:

- QFullSampleSize: Controls how quickly the VMkernel should reduce the queue depth for the LU returning DEVICE BUSY/QUEUE FULL status; by default, QFullSampleSize is set to 0, which disables adaptive queuing
- QFullThreshold: Controls how quickly the queue depth should be restored once congestion has been addressed

vSphere administrators often enable adaptive queuing as a means to address storage congestion issues. However, while this approach can temporarily help reduce storage congestion, it does not address the root cause of the congestion. Moreover, although adaptive queuing can enhance performance during times when storage is congested, overall I/O performance is superior in a well-tuned, balanced configuration.

Before enabling adaptive queuing, HP highly recommends examining your environment to determine the root cause of transient or permanent I/O congestion. For well-understood, transient conditions, adaptive queuing may help you accommodate these transients at a small performance cost. However, to address more permanent I/O congestion, HP offers the following suggestions:

- Using VMware disk shares
- Using VMware Storage IO Control (SIOC)
- Re-evaluating the overall ability of the storage configuration to accommodate the required workload – for example, non-vSphere hosts may have been added to a well-tuned, balanced SAN, placing additional stress on the shared storage; such configuration changes must be investigated and addressed before you throttle back the vSphere hosts.

If you must enable adaptive queuing, HP recommends the following values for HP EVA Storage homogeneous environments:

- QFullSampleSize = 32
- QFullThreshold = 4

Note the recommendation for QFullThreshold is changed from eight (previous recommendation) to four. This will work just as well with EVA and also make EVA array coexistence with 3PAR configurations much more efficient.

For environments where the ESX servers are connected to both HP 3PAR Storage and HP EVA Storage, then the following values are recommended:

- QFullSampleSize = 32
- QFullThreshold = 4

Best practices for adaptive queuing

- Rather than enabling adaptive queuing, determine the root cause of the I/O congestion.
- If you decide to use adaptive queuing in an homogenous HP EVA Storage environment or an EVA/3PAR heterogeneous environment, set the values as follows: QFullSampleSize = 32 and QFullThreshold = 4.

In ESXi releases earlier than ESXi 5.1, QFullSampleSize and QFullThreshold are global settings. The value set will apply to all devices accessible to the ESX server. In ESXi 5.1, because of different recommended optimal values for various arrays, these parameter can now be set more granularly on a per device level.

Run the following esxcli command:

```
esxcli storage core device set --device  device_name --queue-full-threshold  Q --
queue-full-sample-size S
```

Where S is the value for QFullSampleSize and
Q is the value for QFullThreshold

These settings are persistent across reboots and you can confirm their value by access the specific device properties with the command:

```
esxcli storage core device list --device device
```

Using the paravirtualized virtual SCSI driver

First available in vSphere 4, the VMware paravirtualized virtual SCSI (pvSCSI) driver is installed with the VMware Tools. This guest OS driver communicates directly with VMware Virtual Machine Monitor, helping to increase throughput while reducing latency.

In testing, the pvSCSI driver has been shown to increase the performance of Vdisks compared to standard virtual SCSI adapters such as LSILogic and BusLogic.

When using Iometer, for example, there was a performance improvement of between 10% – 40%, depending on the workload used. The tested I/O block sizes were 4 KB, 8 KB, and 64 KB for sequential and random I/O workloads.

Best practices for improving the performance of VMs with I/O-intensive workloads

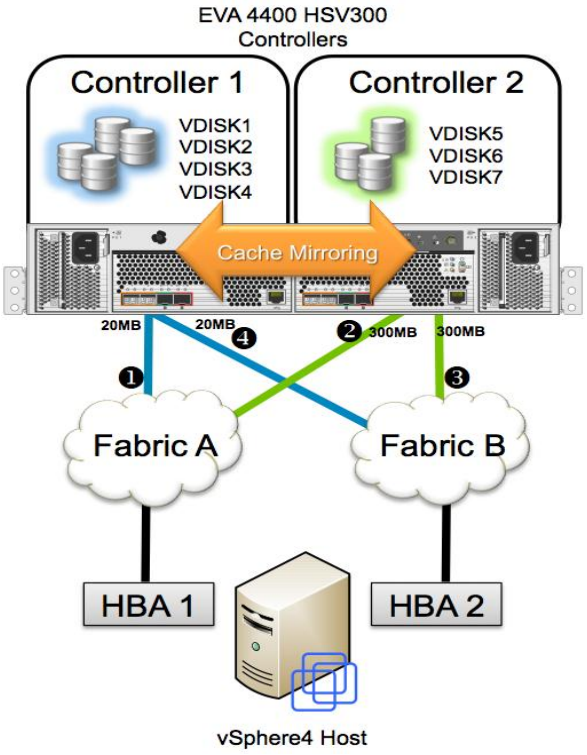
- Consider using the pvSCSI driver with the VM's data logical units, which can enhance performance by 10% – 40%, depending on the particular workload used.

Monitoring EVA performance in order to balance throughput

ALUA compliance in vSphere 4.x has significantly reduced configuration complexity and given you the ability to quickly configure a balanced Vdisk environment. However, you should monitor EVA host port performance to ensure that this configuration is also balanced from the perspective of I/O throughput.

Even though Vdisk access may be well-balanced between the two controllers in a particular configuration, it is possible that most or all I/O requests are going to Vdisks on just one of these controllers. In this scenario, the resources of one of the controllers are not being fully utilized. For example, Figure 21 shows an environment where the majority of I/O throughput is routed through a single controller despite the environment being balanced from the perspective of Vdisk access. Each port on Controller 2 is processing 300 MB/s of I/O throughput, while ports on Controller 1 are only processing 20 MB/s each.

Figure 21. Unbalanced I/O access in a vSphere 4.x environment

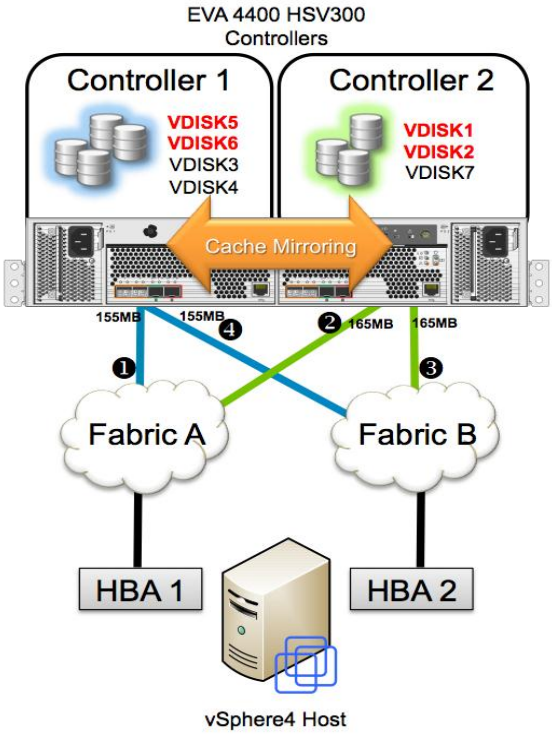


Better balanced throughput could be achieved in this example by moving one or more of the Vdisks on Controller 2 (Vdisk5, 6 or 7) to Controller 1. Simply update the Vdisk controller access preference within Command View EVA to the desired value; alternatively, if you need to move multiple Vdisks to an alternate controller, you could use SSSU scripting tools.

Within a few minutes of the update, vSphere 4.x will switch the I/O access path of the Vdisk to the new controller, resulting in better balanced I/O accesses that may, in turn, lead to improved I/O response times.

Figure 22 shows a better-balanced environment, achieved by moving the controller ownerships of Vdisk 5 and 6 to Controller 1 and of Vdisk1 and 2 to Controller 2.

Figure 22. Balanced I/O access in a vSphere 4.x environment after changing controller ownership for certain Vdisks



This type of tuning can be useful in most environments, helping you achieve the optimal configuration.

Monitoring EVA host port throughput

In order to achieve the balance described above, you must be able to monitor EVA host port throughput. HP recommends using EVAperf, a utility that is bundled with Command View EVA management software and can be accessed from the desktop of the Command View EVA management station.

For more information on using EVAperf to monitor host port throughput, refer to Appendix C: Balancing I/O throughput between controllers.

Best practice for monitoring EVA host port throughput

- To allow you to make proactive adjustments, use EVAperf to monitor EVA performance.

Optimizing I/O size

You can enhance the performance of applications that generate large I/Os by reducing the maximum I/O size that vSphere hosts can send to the EVA array.

By design, vSphere 4.x allows I/Os as large as 32 MB to be sent to the array. You can control I/O size via the vSphere advanced parameter Disk.DiskMaxIOSize. HP recommends setting this value to 128 KB for an EVA array to optimize overall I/O performance.

Note

VMware makes a similar recommendation in their knowledge base article 1003469.

Best practice for improving the performance of VMs that generate large I/Os

- Consider setting the vSphere advanced parameter Disk.DiskMaxIOSize to 128 KB to enhance storage performance.

VMware vStorage API for Array Integration (VAAI)

Starting with firmware 10100000, the HP EVA Storage arrays began supporting VMware VAAI. VMware VAAI provides storage vendors access to a specific set of VMware Storage APIs, which enable offloading specific I/O and VM management operations to the storage array. With hardware offload, an administrator is able to reduce memory and CPU consumption on VAAI enabled ESX/ESXi hosts, reduce storage array port and fabric I/O traffic, and significantly increase performance and scalability of the configuration. Please consult HP support pages for the latest active and supported EVA firmware versions that support VAAI.

VMware VAAI storage integration is made possible by leveraging specific primitives within the API. Each of the primitives enables hardware offload of a specific capability. The following table summarizes the VAAI primitives support by HP EVA Storage

Table 8. VAAI Primitives support

Primitive	Equivalent Name	ESX 4.1/ESXi 4.1	ESXi 5.x
Full Copy/Hardware Assisted Move	XCOPY	YES	YES
Block Zeroing/Hardware Assisted Zeroing	WRITE SAME	YES	YES
Hardware Assisted Locking	ATS	YES	YES
Out of Space Awareness	TP Stun (VM Pause)	YES	YES
Thin Provisioning			
Thin Provisioning Reporting	TP CAPABILITY	NA	YES
Soft Threshold Monitoring	NA	NA	YES
Hard Threshold Monitoring	NA	NA	YES
Space Reclamation	UNMAP	NA	YES*
ESX VAAI Software Plugin required	NA	YES	NO

*UNMAP supported in XCS 1120000 or newer. See space reclamation section below.

Table 8 summarizes VAAI primitives supported with HP EVA Storage for ESX/ESXi 4.1 and ESXi 5. A few important things to note from this summary are:

- VAAI Support

  VMware only supports VAAI on ESXi/ESX 4.1 and ESXi 5. ESX/ESXi 4.0 is not supported. Customers using vSphere 4 will need to upgrade to vSphere 4.1 at minimum to benefit from VAAI offload acceleration. It is recommended to upgrade to ESXi 5.x in order to leverage a much simpler VAAI configuration, because ESXi 5 does not require a VAAI software plugin.

- VAAI Software Plugin

  Each storage vendor provides a VAAI software plugin for their respective arrays. A VAAI plugin is required when using ESX/ESXi 4.1. Some arrays have this software plugin built into ESX and require no asynchronous installation. The HP EVA Storage VAAI plugin is a software plugin that must be asynchronously downloaded and installed on ESX/ESXi 4.1. Note that for HP EVA Storage arrays, a VAAI software plugin is not required for ESXi 5.x.

- Thin Provisioning

  Thin Provisioning support was introduced with ESXi 5 and is not supported on ESX/ESXi 4.1

- TP STUN (VM Pause)

  Despite ESX/ESXi 4.1 not supporting any of the Thin Provisioning features, ESX/ESXi 4.1 does support the ability to pause a VM, on detection of an out of space condition.

- Plugin-less deployment

  The HP EVA Storage firmware 10100000 VAAI implementation is SCSI standards based which is directly compatible with the standards based implementation of VAAI in ESXi 5. Therefore, no VAAI software plugin is required when using ESXi 5 with HP EVA Storage VAAI enabled.

In this paper we will not explain what each primitive does because that subject has been abundantly discussed in other HP EVA Storage documentation such as the HP EVA Storage array release notes. On the other hand, the paper will focus the discussion on important caveats to take into considerations when deploying VAAI with HP EVA Storage while highlighting recommended VAAI deployment best practices

VAAI alignment considerations

Starting with ESXi 5, VMware VMFS partitions created with ESXi 5 are aligned on 1MB boundary of the underlying logical unit address space presented from storage. Therefore, all VMDKs carved from VMFS volumes are also aligned on 1MB boundaries.

On the other hand, on ESX 4.1, VMFS partitions are 64KiB aligned and VMDKs carved from these VMFS volumes in turn are 64KiB aligned. When an ESX/ESXi 4.1 host is upgraded to ESXi 5, datastores that were created under ESX/ESXi 4.1 will retain the alignment they were created with.

Because VAAI WRITE SAME and XCOPY operations are performed respectively to initialize or clone VMDKs, these operations conform to the alignment of the VMFS partition address space. When deploying HP EVA Storage VAAI solutions, VMFS partitions to be used for VAAI acceleration must be 64KiB or 1MB aligned.

Datastore that are not 64KiB or 1MB aligned will not be able to leverage VAAI acceleration benefits and will revert back to using legacy ESX data movers to perform clone and zeroing operations. For this reason, when planning to deploy VAAI with HP EVA Storage, datastores that do not meet this alignment recommendation need to be re-aligned to 64KiB or 1MB alignment. In some cases this may require evicting all the VMs from this datastore, then deleting and recreating the datastore with the desired alignment.

Best practice for aligning VMFS volume for use with HP EVA Storage VAAI

- VMFS datastore must be aligned to either 64KIB or 1MB boundary for use with HP EVA Storage VAAI integration.

VAAI transfer size consideration

VAAI WRITE SAME requests on ESX 4.1 and ESXi 5.x have a default transfer size of 1MB whereas XCOPY requests default to a 4MB transfer size (tunable to as much as 16MB). For maximum performance of VAAI offloading operations to storage, the HP EVA Storage has been optimized for XCOPY and WRITE SAME request that are multiples of 1MB. Transfer sizes that are non-multiple of 1MB, will revert to utilizing ESX legacy data movers. Furthermore, note that VMware VAAI does not support clone operations between datastores of different block sizes. So when using ESX 4.1 with EVA, it is important to ensure that the datastores that are used for VAAI clone operations are either 64k or 1MB aligned and that they are also of the same block size.

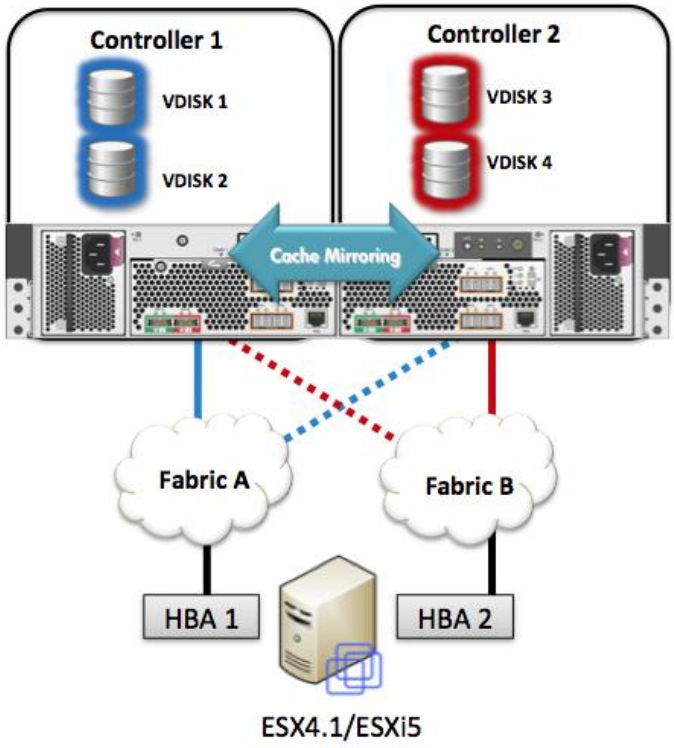
VAAI and EVA proxy operations considerations

As discussed earlier in this paper, HP EVA Storage arrays are asymmetric active-active arrays and ALUA compliant. In this architecture, a logical unit can be accessed through both controllers of the EVA. However, one of the two controllers is an optimal (often referred to as preferred) path to the logical unit. VAAI WRITE SAME and XCOPY primitives will adhere to these controller preference rules as follows:

- WRITE SAME commands to a logical unit will be sent to the controller that is the optimal path to the logical unit.
- XCOPY commands copy data from one logical unit to another (where source and destination are allowed to be the same). During XCOPY operations, XCOPY commands are sent to the controller that owns the destination logical unit.

Figure 23 below illustrate an HP EVA Storage configuration where controller 1 owns Vdisks 1 and 2 and controller 2 owns Vdisks 3 and 4.

Figure 23. VAAI command pathing and optimal path access



In this configuration, access to the Vdisks by VAAI WRITE SAME and XCOPY commands is summarized in the table below:

Table 9. VAAI command controller access

Command	Operation	Controller Receiving Command
WRITE SAME	Zero addresses on Vdisk1 or Vdisk2	Controller 1
WRITE SAME	Zero addresses on Vdisk3 or Vdisk4	Controller 2
XCOPY	Copy data from Vdisk1 to Vdisk1	Controller 1
XCOPY	Copy data from Vdisk1 to Vdisk2	Controller 1
XCOPY	Copy data from Vdisk1 to Vdisk3 or Vdisk4	Controller 2
XCOPY	Copy data from Vdisk3 to Vdisk3	Controller 2
XCOPY	Copy data from Vdisk3 to Vdisk4	Controller 2
XCOPY	Copy data from Vdisk3 to Vdisk1 or Vdisk2	Controller 1

From the summary in Table 9, we can see that WRITE SAME operations are sent to the optimal “owning” controller for a Vdisk. So when zeroing multiple Vdisks concurrently, there is a significant performance advantage of zeroing Vdisks owned by different controllers concurrently over Vdisks that are owned by the same controller.

Similarly, during XCOPY operations, EVA controllers work together to transfer the data from the source Vdisk to the destination Vdisk using the controller proxy link when the source and destination Vdisks are on different controllers. Hence for improved performance it is best to perform XCOPY operations between Vdisks that are owned by different controllers.

Best practice for zeroing

- For improved performance, when performing zeroing activities on two or more Vdisks concurrently, it is best to have the Vdisks ownership evenly spread across controllers.

Best practice for XCOPY

- XCOPY operations can benefit from increased performance when different EVA controllers own the source and destination datastores.
- When an HP EVA Storage array is in degraded mode, such as when one of the two controllers is down, the proxy relationship between the two controllers will also be affected. Under this condition, only the remaining controller will process all XCOPY and zeroing operations in addition to servicing non-VAAI I/Os. Therefore, when the EVA is in a degraded mode, VAAI operations should be avoided.

Best practice for degraded EVA

- When the EVA is in degraded mode, avoid using VAAI operations.

VAAI and Business Copy and Continuous Access

HP EVA Storage local and remote replication of a Vdisk generates additional workloads for the diskgroups servicing this Vdisk. The EVA VAAI implementation has carefully taken into consideration the tradeoffs that exist between speeding up VAAI workloads versus equally servicing other important non-VAAI workloads without performance impact. Therefore, when using VAAI primitives on a Vdisk that is in a Business Copy or Continuous Access relationship, administrators should expect throttled VAAI performance gains.

Best practice for Business Copy and Continuous Access

- When using VAAI on Vdisks that are in snapshot, snapclone, mirror clone, or continuous access relationships VAAI performance may be throttled.

VAAI operation concurrency

HP EVA Storage will allow VAAI cloning and zeroing operations or any combination of them to be performed concurrently and in parallel to other VAAI and non-VAAI operations. However, as previously discussed, the HP EVA Storage array VAAI implementation was also carefully designed to only allow VAAI operation a finite amount of system resources in order to reduce the performance impact on external and internal non-VAAI I/O workloads. This is a design choice that guarantees that VAAI clone/zeroing operations adequately share storage array resources and bandwidth.

Following the VAAI best practices discussed in this guide will provide the best performance to VAAI clone and zeroing operations while maintaining adequate I/O latencies and throughput for VAAI and non-VAAI workloads.

VAAI clone and zeroing operations offer the benefit of increased performance to VM cloning, VM template creation and VM initialization. However, queuing multiple concurrent such operations may induce undesired latency to the overall system affecting the performance of VAAI and non-VAAI workloads. Therefore, it is best to keep the number of concurrent VAAI operations to a minimum, which will help keep VAAI performance high and lower the performance impact on non-VAAI workloads.

Best practice for concurrent VAAI operations

- Minimize the number of concurrent VAAI clone and/or zeroing operations to reduce the impact on overall system performance.
- An inadequate number of physical disk spindles in a disk array can bottleneck the performance of any disk subsystem despite cache sizes, controller count or speed. To benefit from the performance advantages of VAAI and concurrent VAAI operations, it is critical that the deployed HP EVA Storage array is configured with an adequate minimum number of hard drives. The following table provides minimum estimates for the number of drives needed to achieve adequate VAAI copy rates with Vraid5 Vdisks. Note that these estimates are dependent on the overall I/O workload of the system.

Table 10 provides the number of drives to use with VAAI.

Table 10. VAAI command controller access

HP EVA Storage	7.2K RPM HDD	10K RPM HDD	15K RPM HDD
EVA4400	*13	65	55
EVA6400	110	60	45
EVA8400	110	60	45
P6300/P6350	140	75	55
P6500/P6550	175	90	65

Best practice for minimum drive requirements for VAAI

- Ensure the HP EVA Storage is configured with an adequate number of drives when using VAAI. Refer to Table 10 for guidance.

VMware Space Reclamation and EVA

As discussed in the VAAI section above, VMware ESX added support for Space Reclamation in ESXi 5.0. Support of the UNMAP VAAI primitives enables storage arrays that implement the UNMAP SCSI command to successfully process space reclamation requests. Starting with firmware 11200000, the HP EVA Storage arrays support capacity reclamation as support for the UNMAP SCSI primitive has been added. Table 11 shows the HP EVA Storage arrays that support UNMAP.

Table 11. HP EVA Storage arrays supporting UNMAP

HP EVA Storage
EVA4400
EVA6400
EVA8400
P6300/P6350
P6500/P6550

ESX UNMAP support history

Though UNMAP support was added in ESXi 5.0, support for Space Reclamation in ESXi 5.0 was retracted a little while after ESXi 5.0 released. The performance of UNMAP operations at the storage were directly tied to the success of Virtual Infrastructure (VI) operations such as, Storage VMotion and VMware VM Snapshots. The longer a storage system took to process UNMAP commands, the longer such VI operations would take and cause them to effectively timeout. Because of the disparity in performance observed by various vendors' storage arrays with respect to the processing of UNMAP commands, VMware retracted support for Space Reclamation in ESXi 5.0.

ESXi 5.0 patch2 was released soon after ESXi 5.0 and disabled the UNMAP functionality in ESXi. When UNMAP support was re-introduced in ESXi 5.0U1, ESX UNMAP handling was altered to decouple the space reclamation operation from the completion of VI operations such as Storage VMotion and VM Snapshots. These operations would no longer automatically trigger ESX to send UNMAP requests to storage. Instead, a user at their own discretion would have to manually invoke, via command line, the utility that will reclaim available capacity independently of VI operations.

The newly introduced ESXi 5.5 also requires users to manually reclaim capacity at their discretion by invoking a command line utility. However, ESXi 5.5 made many significant improvements over ESXi 5.0U1 that are discussed later in this document.

*13 EVA4400 only supports a maximum of 96 drives. Hence 7.2K RPM drives can't meet the minimum number required for adequate VAAI performance.

Table 12 shows the ESXi UNMAP recommendation usage.

Table 12. ESXi UNMAP usage recommendations

ESX Version	UNMAP Support	Execution Recommendation
ESXi 5.0	Not Supported	NA
ESXi 5.0 patch 2	Not Supported	NA
ESXi 5.0U1	Supported	Offline
ESXi 5.1	Supported	Offline
ESXi 5.5	Supported	Offline

In this paper, we will not explain in detail how the various versions of ESX perform UNMAP. This topic has been abundantly discussed in other documents, papers and VMworld presentations. However we will contrast key features and highlight important considerations when using UNMAP with EVA Storage arrays.

UNMAP in ESXi 5.0Ux and ESXi 5.x

There are many important considerations to keep in mind when performing space reclamation operations. However most considerations can be summed up as follows:

- Reclaim efficiency consideration
- Capacity reservation consideration
- Performance consideration

In ESXi 5.0Ux and ESXi 5.1Ux, capacity is reclaimed by invoking the vmkfstools –y command with a percentage of free capacity as the parameter. vmkfstools –y 50 for example, will attempt to reclaim 50% of the free capacity on the datastore. However for a datastore that has areas of the disk that were never written, then the reclaim request with the 50% flag may result in sending UNMAP requests to physical addresses that were never allocated. Thus leading to no visible capacity returned to the user. Furthermore, it may take multiple reclaim iterations when using a small reclaim flag value before all of the desired capacity to reclaim is effectively reclaimed. For this reason, users tend to want to reclaim with the 99% flag (max supported) instead to guarantee effectiveness.

Reclaiming 99% of free capacity while the datastore is online however makes it subject to capacity reservations and also performance issues. During the reclaim process, 99% of the free capacity on the datastore is exclusively reserved for access by the UNMAP operation. While the capacity to be reclaimed is reserved, existing thin VMDKs that need capacity to grow or new VMDKs that need to be created will fail to allocate space. Making use of the vmkfstools command online is not a very desirable thing especially when factoring in the performance impact induced to the datastore from the UNMAP requests.

These considerations make space reclaiming on these versions of ESX more suitable during maintenance windows, which essentially will require migrating running VMs to another datastore before reclaiming capacity. But it probably is much simpler at that point to delete the volume at the array and recreate it.

When performing space reclamation offline (during maintenance window) on the EVA, it is recommended to use the 99% flag with the vmkfstools –y command. Not only is this flag value more efficient and effective from an ESX point of view, it will also allow the EVA to achieve the highest reclaim efficiency because larger contiguous blocks will be reclaimed.

Best practice for space reclamation with ESXi 5.0U1 and ESXi 5.1

- Space reclamation in ESXi 5.0U1 and ESXi 5.1 is recommend to be performed during a maintenance window.

Note

The EVA Vdisks are striped across all spindles in the array. Accordingly, putting a host in maintenance mode to carry out the reclaim operation on inactive ESX datastores (EVA Vdisks) can still have an impact on volumes that are active on other hosts because all volumes share the same spindle.

Best practice when only 80% of expected capacity is being reclaimed

- Set the Vdisk access mode to write-through and immediately set it back to write-back and retry the reclaim operation.

Finally, it is recommended to run only one instance of VMware block space reclamation operation at a time per EVA array and per datastore.

UNMAP in ESXi 5.5

In ESXi 5.5, the same considerations discussed above apply. However UNMAP in ESXi 5.5 is performed using the `esxcli –u datastore_uuid –n reclaim_unit` instead of the vmkfstools.

Note

The vmkfstools will work with this version of ESX; however, it is deprecated and the syntax is different.

This command reclaims capacity in increments specified by the reclaim_unit. The reclaim_unit defaults to 200 blocks. Thus on a 1MB datastore, it will attempt to reclaim capacity in 200MB chunks.

By reclaiming capacity in much smaller chunks, ESXi 5.5 is able to effectively address the capacity reservation issue that previous versions were subject to.

Additionally, ESXi 5.5 always reclaims 100% of the free capacity instead of some arbitrary percentage and only a single iteration of the command is required to achieve this result. Making ESXi 5.5 space reclamation 100% efficient and effective compared to previous versions of ESX.

In order to maximize the EVA's capability to handle UNMAP commands, it is recommended that the following reclaim increments be used with ESXi 5.5:

- For arrays with FC backend - EVA4400 / EVA8400, the recommended value of reclaim unit is 480.
- For arrays with SAS backend - P63xx / P65xx, the recommended value of reclaim unit is 1280.

Best practice for space reclamation with ESXi 5.5

- Space reclamation in ESXi 5.5 needs to be performed offline.

Best practice for ESXi 5.5 reclaim unit for EVA (assuming 1MB block size)

- Use reclaim unit of 480 for EVA4400/EVA8400.
- Use reclaim unit of 1280 for P63xx/P65xx.

Table 13 provides an overview of key differences between the operation of UNMAP in the versions that support UNMAP.

Table 13. Differences in versions supporting UNMAP

Feature	ESXi 5.0U1/ESXi 5.1	ESXi 5.5
UNMAP invoked via	Command Line	Command Line
CLI	vmkfstools	esxcli
Uses temp file	YES	YES
Max temp file size	2TB	~62TB
Efficiency	Random	100%
Default reclaim unit (1MB Block Size)	60% of free capacity	200 Blocks
Recommended reclaim unit (1MB Block Size)	-	480 (EVA4400/EVA8400) 1280 (P63xx/P65xx)
Capacity Reservation	HIGH	LOW
Maintenance Window required	YES	NO
Support multiple UNMAP descriptors	NO	YES

UNMAP alignment, size and EVA host mode considerations

In order to optimize space reclamation efficiency and performance, ESX's UNMAP implementation is adaptive. The implementation leverages the data retrieved from the array SCSI inquiry page B0 (Blocks Limits page) and adjusts ESX behavior to match the array expected values. Two critical values retrieved in the blocks limits inquiry page are:

- OPTIMAL UNMAP GRANULARITY
- MAXIMUM UNMAP BLOCK COUNT

The EVA advertises an UNMAP optimal unmap granularity that nicely aligns with ESX expected default alignments of 64KiB and 1MB. Hence, VMware VMFS partitions created with ESXi 5 and aligned on 1MB or upgraded VMFS datastores which retain their alignment and block size (which typically defaults at 64k) will all function properly with UNMAP on EVA. Note however that ESX will not issue UNMAP requests to VMFS partitions aligned at offsets other than 64KiB and 1MB on an EVA. Therefore, it is recommended to have all EVA datastores aligned at 64KiB or 1MB in order to use space reclamation.

Best practice for aligning VMFS volume for use with HP EVA Storage UNMAP

- VMFS datastore must be aligned to either 64KiB or 1MB boundary for use with HP EVA Storage UNMAP integration.

The maximum unmap block count advertises to ESX the maximum unmap request size the array will allow. Requests above this value will be rejected and have to be retried. Since this value is retrieved directly from the storage system by ESX, no actions are required by the user to leverage this functionality when using ESX.

In order for ESX to appropriately adapt to the EVA desired UNMAP behavior, it is critical that the ESX hosts accessing the EVA be set to the VMware host mode in Command View EVA. Failure to do so will lead to unpredictable results.

Note that when RDM volumes are presented to VMs running operating systems that support space reclamation natively, such as Windows Server 2012, space reclamation within these VMs is not supported. This is because the RDM will be accessed by ESX using the VMware host mode, and that host mode is not suitable for space reclamation on Windows Server 2012. If you require a virtual machine to perform space reclamation on EVA, at the time of this writing, you must configure VMDirectPathIO and configure a host with the appropriate host mode for the operating system the virtual machine is running.

Best practice for host mode configuration

- The VMware host mode must be used on the EVA for proper operation of the UNMAP primitive with ESX.
- UNMAP is not supported by VMs accessing RDM volumes presented to ESX hosts. Use, VMDirectPathIO and give exclusive access of the volume to the VM with appropriate host mode.

Table 14 shows the differences between UNMAP operation in ESXi 5.0 and ESXi 5.0U1 and later.

Table 14. Differences in UNMAP operation in ESXI 5 versions

ESXi 5.0	ESXi 5.0U1 and later
Single automated operation	Manual Step 1
VM/VMDK is deleted	VM/VMDK is deleted
Used blocks are freed in VMFS	Used blocks are freed in VMFS
ESX sends UNMAP requests to storage	Manual Step 2
Storage frees physical blocks	Space reclamation utility invoked
	VMFS creates a temp file for capacity allocation
	ESX sends UNMAP to the temp file block range
	Storage frees physical blocks

Summary of best practices

How can I best configure my storage?

- To size an EVA disk group, start by understanding the characteristics of the application's workload to ensure the virtualized environment is capable of delivering sufficient performance.
- Using a single EVA disk group is typically sufficient to satisfy all your storage optimization objectives (cost, performance, and capacity).
- Fill the EVA with as many disks as possible, using the largest, equal-capacity disks.
- The single-drive protection level should be sufficient for your disk group unless MTTR is longer than seven days.
- Do not use Vraid6 unless absolutely necessary. Vraid5 is less expensive and provides adequate redundancy for a vSphere deployment.
- When using disks with differing performance characteristics, use a single disk group rather than multiple disk groups each containing disks of the same characteristics.
- Alternate controller ownership for EVA Vdisks between Controller A and Controller B using the Path-A-Failover/Failback or Path-B-Failover/Failback setting.
- Ensure that the managing controllers for DR groups are spread between Controller A and Controller B to maintain an adequate balance in the system configuration.
- The VMware host mode must be used on the EVA for proper operation of the UNMAP primitive with ESX.
- UNMAP is not supported by VMs accessing RDM volumes presented to ESX hosts. Use, VMDirectPathIO or NPIV and give exclusive access of the volume to the VM with appropriate host mode.
- VMFS datastore must be aligned to either 64KiB or 1MB boundary for use with HP EVA Storage UNMAP integration.

Which is the best I/O path policy to use for my storage?

- Round robin I/O path policy is the recommended setting for EVA active-active arrays except for Vdisks used by an MSCS cluster, which should be set to MRU.

  MRU is also suitable for other applications if round robin is undesirable in a specific environment.
- Avoid using Fixed I/O path policy with vSphere 4.x/5.x and an EVA array.
- Configure round robin advanced parameters to IOPS=1 for vSphere 4.x/5.x.
- For normal production I/O, avoid using Fixed_AP I/O path policy with vSphere 4.1 or its equivalent Fixed I/O path policy with vSphere 5.x and an EVA array.

  However, you can leverage Fixed_AP I/O path policy (vSphere 4.1) and Fixed I/O path policy (vSphere 5.x) to quickly rebalance the Vdisk configuration after a controller has been restored from failure. Use just one vSphere 4.1/5x server from the cluster; ensure the array is not under heavy load.

How do I simplify storage management, even in a complex environment with multiple storage systems?

- Use the Storage Module for vCenter to save time and improve efficiency by mapping, monitoring, provisioning, and troubleshooting EVA storage directly from vCenter.
- In multi-vendor, ALUA-compliant SAN environment, you should select the default PSP for the VMW_SATP_ALUA based on the PSP that has been recommended for one of the following:
  - The most prevalent array in the SAN
  - The array with the most Vdisks provisioned for vSphere access
- In an exclusively EVA environment, change the default PSP option for VMW_SATP_ALUA to VMW_PSP_RR.
- Round robin I/O path policy is recommended for EVA active-active arrays. MRU is also suitable if round robin is not desired in a particular environment.
- Configure MSCS cluster Vdisks to use MRU I/O path policy. Since the recommended default setting for all EVA Vdisks is round robin, you must manually configure MSCS Vdisks to MRU.
- Alternate the controller ownership of EVA Vdisks between Controller A and Controller B by configuring the Path-A-Failover/Failback or Path-B-Failover/Failback setting.
- With vSphere 4.1/5x, create a new SATP rule for each storage system.
- Use the same Vdisk ID for all vSphere servers in the same cluster.
- Use a simplified name for your datastores and be consistent, thus accommodating vSphere hosts you may later add to the cluster.
- To properly align the filesystem, use the vSphere client when creating a datastore.
- When naming a datastore, use the same name you selected in Command View EVA when creating the logical unit/Vdisk.
- Utilize HP Insight Control Storage Module for vCenter to save time and improve efficiency by mapping, monitoring, provisioning, and troubleshooting EVA storage directly from vCenter.
- When all GbE ports are connected to the iSCSI SAN and two NICs are used at the host, use Static discovery to control the host access to desired targets when using a 1GbE iSCSI Module option.
- To simplify configuration of the 1GbE iSCSI Module and meet adequate high availability, use Dynamic discovery at the ESX host when only two NICs are accessing the iSCSI SAN and up to two GbE ports are used per controller.
- When all 10GbE ports are connected to the iSCSI SAN and two NICs are used at the host, use Dynamic discovery to simply configuration and meet adequate high availability with a 10GbE iSCSI Module option.
- Space reclamation in ESXi 5.0U1, ESXi 5.1 and ESXi 5.5 is recommend to be performed during a maintenance window.
- Set the Vdisk access mode to write-through and immediately set it back to write-back and retry the reclaim operation when the expected reclaimed capacity is consistently less than 80% and when the 99% reclaim flag is used.

How can I best monitor and tune the EVA array in order to optimize performance?

- If increasing the queue depth at the HBA level, also increase the value of the vSphere advanced parameter Disk.SchedNumReqOutstanding.
- When using VMs with I/O-intensive workloads, consider using paravirtualized virtual adapters for the VM's data logical units, which can provide a 10% – 40% performance improvement, depending on the workload.
- Unless using Windows Vista, Windows 7, or Windows Server 2008, ensure that data drives within the guest OS are properly aligned.
- When using VMs that generate large-sized I/Os, consider setting the vSphere 4.x/5.x advanced parameter Disk.DiskMaxIOSize to 128 KB to increase storage performance.
- HP recommends leaving QFullSampleSize and QFullThreshold at their default – disabled – values and, instead, investigating the root cause of any I/O congestion. If you do choose to enable adaptive queuing, HP recommends the following settings:
  - QFullSampleSize = 32
  - QFullThreshold = 4
- Use EVAperf to monitor EVA performance in order to make proactive adjustments.
- VMFS datastore must be aligned to either a 64KiB or 1MB boundary for use with HP EVA Storage VAAI integration.
- For improved performance, when performing zeroing activities on two or more Vdisks concurrently, it is best to have the Vdisks ownership evenly spread across controllers.
- XCOPY operations can benefit from increased performance when different EVA controllers own the source and destination datastores.

- When the EVA is in degraded mode, avoid using VAAI operations.
- When using VAAI on Vdisks that are in snapshot, snapclone, mirror clone or continuous access relationships VAAI performance may be throttled.
- Minimize the number of concurrent VAAI clone and/or zeroing operations to reduce the impact on overall system performance.
- Ensure the HP EVA Storage is configured with an adequate number of drives when using VAAI. Refer to Table 8 for guidance.
- Using two iSCSI GbE ports at the array and two NICs on the ESX host provides the most flexible option to achieve the right balance of ease of configuration, high availability and performance for the 10GbE iSCSI Module configuration options.
- For the 1GbE iSCSI Module configuration option, using Static discovery to limit the number of iSCSI targets used per GbE port to 1 can help achieve increased high availability and performance at the cost of ease of configuration.
- Use ESXi 5.5 UNMAP reclaim unit of 480 for EVA4400/EVA8400 (assumes 1MB block size).
- Use ESXi 5.5 UNMAP reclaim unit of 1280 for P63xx/P65xx (assumes 1MB block size).
- Invoke only a single instance of VMware space reclamation against an array or datastore.

How do I maintain the availability of Command View EVA deployed in a VM?

- HP recommends deploying Command View EVA (CV EVA) on the local datastore of a vSphere server. However, if a SAN-based deployment s required, then load CV EVA on to multiple EVAs to ensure that the management interface remains available. If CVA EVA were deployed on a single VM, the management interface would be lost if the particular EVA became inaccessible.

Summary

In most environments, the best practices highlighted in this document can help you reduce configuration time and improve storage performance. However, as with all best practices, you must carefully evaluate the pros and cons of the recommendations presented herein and assess their value in your particular environment.

In addition to serving as a reference guide for anyone configuring an EVA-based SAN in conjunction with vSphere 4.x/5.x, this document also provides valuable information about the latest VMware technologies, such as the multi-pathing storage stack.

Glossary

Term	Description
1 GbE iSCSI Module	Built-in 1 GbE iSCSI Module for HP EVA P6000. Also referred to as iSCSI Module
10 GbE iSCSI Module	Built-in 10 GbE iSCSI Module for HP EVA P6000. Also referred to as iSCSI/FCoE Module
Array	In the context of this document, an array is a group of disks that is housed in one or more disk enclosures. The disks are connected to two controllers running software that presents disk storage capacity as one or more virtual disks. The term “array” is synonymous with storage array, storage system, and virtual array.
Controller firmware	The firmware running on each controller within the array manages all aspects of array operation, including communications with Command View EVA.
Default disk group	The default disk group is the disk group created when the array is initialized. This group must contain a minimum of eight disks, with its maximum size being the number of installed disks.
Disk group	A disk group is a named group of disks that have been selected from disks that are available within the array. One or more virtual disks can be created from a disk group.
DR group	A data replication (DR) group is a logical group of virtual disks that is part of a remote replication relationship with a corresponding group on another array.
ESX/ESXi	ESX/ESXi is the hypervisor component of VMware vSphere.

Term	Description
EVA	The HP Enterprise Virtual Array (EVA) Storage product allows pooled disk capacity to be presented to hosts in the form of one or more variably-sized physical devices. The EVA consists of disks, controllers, cables, power supplies, and controller firmware. An EVA may be thought of as a storage system, a virtual array, or a storage array. In this paper the term EVA is also used to refer to classic HP EVA4400/6400/8400 Storage products and the HP EVA P6000 Storage.
Failover	The failover process causes one controller to assume the workload that was previously running on a failed, companion controller. Failover continues until the failed controller is once again operational.
Host	A host is a computer that runs user applications and uses information stored on an array.
LU	The logical unit (LU) is a SCSI convention used to identify elements of a storage system; for example, hosts see a virtual disk as an LU. An LU is also referred to as a Vdisk. The logical unit number (LUN) assigned by the user to a Vdisk for a particular host is the LUN at which that host will see the virtual disk.
LUN	The logical unit number (LUN) is a SCSI convention used to enumerate LU elements; for example, the host recognizes a particular Vdisk by its assigned LUN.
Management server	A management server runs management software such as HP Command View EVA and HP Replication Solutions Manager.
RDM	VMware Raw Device Mapping (RDM) technology allows an LU to be mapped directly from an array to a VM.
SAN	A Storage Area Network (SAN) is a network of storage devices that includes the initiators required to store information on and retrieve information from these devices; the SAN includes a communications infrastructure.
Server-based management	The term “server-based management” implies management from a server.
SSSU	The Storage System Scripting Utility (SSSU) is an HP command-line interface that can be used to configure and control EVA arrays.
Storage area network	See “SAN”.
Storage array	Generic term for an HP EVA Storage product.
Storage system	Generic term for an HP EVA Storage product.
Storage System Scripting Utility	See “SSSU”.
UNMAP	SCSI command used for space reclamation.
UUID	The Unique Universal Identifier (UUID) is a unique, 128-bit identifier for each component of an array. UUIDs are internal system values that cannot be modified by the user.
Vdisk	See “virtual disk”.
Virtual array	Generic term for an EVA; see also “virtual disk”.
Virtual disk	A virtual disk provides variable disk capacity that is defined and managed by the array controller. This capacity is presented to hosts as a disk. A virtual disk may be called a Vdisk in the user interface.
VM	A virtual machine is a guest operating system that runs on a vSphere (ESX) host.
Vraid	Vraid is an EVA representation of RAID levels.
WWNN	World Wide Node Name
WWPN	World Wide Port Name

Appendix A: Using SSSU to configure the EVA

The sample SSSU script provided in this appendix creates and presents multiple Vdisks to vSphere hosts.

The script performs the following actions:

- Create a disk group with 24 disks.
- Set the disk group sparing policy to single-drive failure.
- Create Vdisk folders.
- Add two vSphere hosts and their respective HBA presentation to the EVA and assign the appropriate host profiles.
- Create five Vdisks and present each to both vSphere hosts.

Sample script

```
!
! SSSU CAPTURE script checksum start
!
! CAPTURE CONFIGURATION Step1A on Sun Oct 11 18:02:46 2009
!
! Manager: localhost
! System: LE_TOP
!
! SSSU Build 012309A for EVA Version 9.0.0
!
SET OPTIONS ON_ERROR=HALT_ON_ERROR
ADD SYSTEM "LE_TOP" DEVICE_COUNT=24 SPARE_POLICY=SINGLE DISKGROUP_DISKTYPE=ONLINE DISKGROUP_TYPE=BASIC
SELECT SYSTEM "LE_TOP"
!
!
! Original controller names were:
!
! \Hardware\Controller Enclosure\Controller 2
! \Hardware\Controller Enclosure\Controller 1
!
!
ADD DISK_GROUP "\Disk Groups\Default Disk Group" DEVICE_COUNT=24 SPARE_POLICY=SINGLE DISKGROUP_DISKTYPE=ONLINE
DISKGROUP_TYPE=BASIX OCCUPANCY_ALARM=100
SET DISK_GROUP "\Disk Groups\Default Disk Group" NAME="DG1"
ADD FOLDER "\Virtual Disks\VM_BOOT_DISKS"
ADD FOLDER "\Virtual Disks\VM_DATA_DISKS"
ADD HOST "\Hosts\ESX1" IP=DYNAMIC_IP_ASSIGNMENT OPERATING_SYSTEM=VMWARE WORLD_WIDE_NAME=2100-0010-8602-003C
SET HOST "\Hosts\ESX1" ADD_WORLD_WIDE_NAME=2100-0010-8602-003D
ADD HOST "\Hosts\ESX2" IP=DYNAMIC_IP_ASSIGNMENT OPERATING_SYSTEM=VMWARE WORLD_WIDE_NAME=2100-0010-8601-EC1C
SET HOST "\Hosts\ESX2" ADD_WORLD_WIDE_NAME=2100-0010-8601-EC1D
ADD VDISK "\Virtual Disks\DATA_DISKS\Vdisk001" DISK_GROUP="\Disk Groups\DG1" SIZE=180 REDUNDANCY=VRAID5
WRITECACHE=WRITEBACK MIRRORCACHE=MIRRORED READ_CACHE NOWRITE_PROTECT OS_UNIT_ID=0 PREFERRED_PATH=PATH_A_BOTH
WAIT_FOR_COMPLETION
ADD Vdisk 1 VDISK="\Virtual Disks\DATA_DISKS\Vdisk001\ACTIVE" HOST="\Hosts\ESX1"
ADD Vdisk 1 VDISK="\Virtual Disks\DATA_DISKS\Vdisk001\ACTIVE" HOST="\Hosts\ESX2"
ADD VDISK "\Virtual Disks\DATA_DISKS\Vdisk002" DISK_GROUP="\Disk Groups\DG1" SIZE=180 REDUNDANCY=VRAID5
WRITECACHE=WRITEBACK MIRRORCACHE=MIRRORED READ_CACHE NOWRITE_PROTECT OS_UNIT_ID=0 PREFERRED_PATH=PATH_B_BOTH
WAIT_FOR_COMPLETION
ADD Vdisk 2 VDISK="\Virtual Disks\DATA_DISKS\Vdisk002\ACTIVE" HOST="\Hosts\ESX1"
ADD Vdisk 2 VDISK="\Virtual Disks\DATA_DISKS\Vdisk002\ACTIVE" HOST="\Hosts\ESX2"
ADD VDISK "\Virtual Disks\DATA_DISKS\Vdisk003" DISK_GROUP="\Disk Groups\DG1" SIZE=180 REDUNDANCY=VRAID5
WRITECACHE=WRITEBACK MIRRORCACHE=MIRRORED READ_CACHE NOWRITE_PROTECT OS_UNIT_ID=0 PREFERRED_PATH=PATH_A_BOTH
WAIT_FOR_COMPLETION
ADD Vdisk 3 VDISK="\Virtual Disks\DATA_DISKS\Vdisk003\ACTIVE" HOST="\Hosts\ESX1"
ADD Vdisk 3 VDISK="\Virtual Disks\DATA_DISKS\Vdisk003\ACTIVE" HOST="\Hosts\ESX2"
ADD VDISK "\Virtual Disks\DATA_DISKS\Vdisk004" DISK_GROUP="\Disk Groups\DG1" SIZE=180 REDUNDANCY=VRAID5
WRITECACHE=WRITEBACK MIRRORCACHE=MIRRORED READ_CACHE NOWRITE_PROTECT OS_UNIT_ID=0 PREFERRED_PATH=PATH_B_BOTH
WAIT_FOR_COMPLETION
ADD Vdisk 4 VDISK="\Virtual Disks\DATA_DISKS\Vdisk004\ACTIVE" HOST="\Hosts\ESX1"
ADD Vdisk 4 VDISK="\Virtual Disks\DATA_DISKS\Vdisk004\ACTIVE" HOST="\Hosts\ESX2"
```

More information

For more information on the SSSU command set, refer to the SSSU user guide, which can be found in the document folder for the Command View EVA install media.

Appendix B: Miscellaneous scripts/commands

This appendix provides scripts/utilities/commands for the following actions:

- Change the default PSP for VMW_SATP_ALUA.
- Set the I/O path policy and attributes for each Vdisk.
- Configure the disk SCSI timeout for Windows and Linux guests.

Changing the default PSP

This command changes the default PSP for VMW_SATP_ALUA:

For ESX 4.x:

```
esxcli nmp satp setdefaultpsp -s VMW_SATP_ALUA -P VMW_PSP_RR
```

For ESXi 5.x:

```
esxcli storage nmp satp set -s VMW_SATP_ALUA -P VMW_PSP_RR
```

Setting the I/O path policy and attributes[14]

This script automatically sets the I/O path policy to round robin and also sets the I/O path attributes for each Vdisk:

Note
This script should only be used for environments with EVA Vdisks connected to vSphere 4.x/5.x servers.

For ESX 4.x

```
for i in 'esxcli nmp device list | grep ^naa.600';do esxcli nmp device setpolicy
-P VMW_PSP_RR -d $i;esxcli nmp roundrobin setconfig -t iops -I 1 -d $i;  done
```

For ESXi 5.x

```
for i in 'esxcli storage nmp device list  | grep ^naa.600';do esxcli storage nmp
device set -P VMW_PSP_RR -d $i;esxcli storage nmp psp roundrobin deviceconfig set
-t iops -I 1 -d $i;done
```

Configuring the disk SCSI timeout for Windows and Linux guests

Change the disk SCSI timeout setting to 60 seconds.

Windows guest
For a VM running Windows Server 2003[15] or earlier, change the value of the HKEY_LOCAL_MACHINE/SYSTEM/CurrentControlSet/Services/Disk/TimeoutValue registry setting to **3c** (that is, **60** expressed in hexadecimal form).

A reboot is required for this change to take effect.

Linux guest
Use one of the following commands to verify that the SCSI disk timeout has been set to a minimum of 60 seconds:

```
cat /sys/bus/scsi/devices/W:X:Y:Z/timeout
```

or

```
cat /sys/block/sdX/device/timeout
```

[14] When cutting and pasting the scripts referenced beware of hidden characters that may cause the scripts to fail. Allows copy in a text editor that will strip hidden characters before running this script.
[15] In Windows Server 2008, the SCSI timeout defaults to 60 seconds.

If required, set the value to **60** using one of the following commands:

```
echo 60 > /sys/bus/scsi/devices/W:X:Y:Z
```

or

```
echo 60 | cat /sys/block/sdX/device/timeout
```

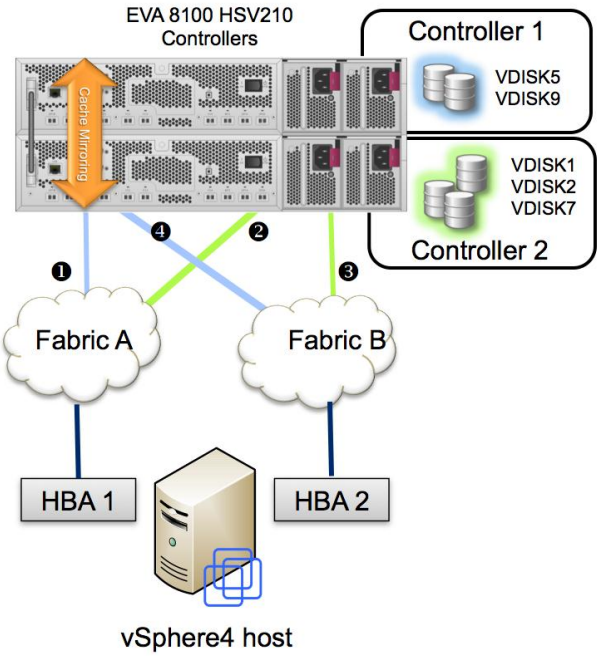
where **W:X:Y:Z** or **sdX** is the desired device.

No reboot is required for these changes to take effect.

Appendix C: Balancing I/O throughput between controllers

The example described here is based on an environment (shown in Figure C-1) with balanced Vdisk access but imbalanced I/O access. The appendix explores the steps taken to balance I/O access.

Figure C-1. Sample vSphere 4.x/5.x environment featuring an HP 8100 Enterprise Virtual Array with two four-port HSV210 controllers



Vdisks are balanced, as recommended in this document, with two Vdisks owned by Controller 1 and three by Controller 2; however, you must also ensure that I/Os to the controllers are balanced. Begin by using the EVAperf utility to monitor performance statistics for the EVA array.

Run the following command:

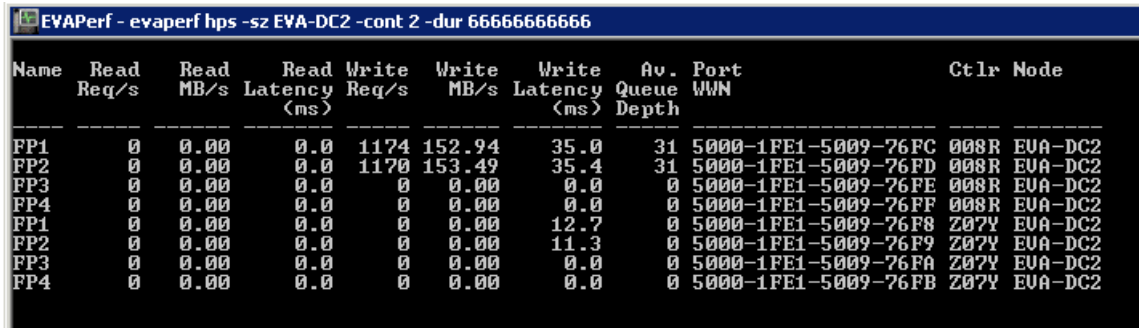
```
evaperf hps -sz <array_name> -cont X -dur Y
```

where X is the refresh rate (in seconds) for statistics and Y is the length of time (in seconds) over which statistics are captured.

Figure C-2 provides sample statistics.

Note
The statics shown in Figure C-2 are not representative of actual EVA performance and can only be used in the context of the example provided in this appendix, which is intended to illustrate the benefits of round robin I/O path policy and ALUA-compliance rather than presenting actual performance.

Figure C-2. I/O routes



In this example, even though the EVA array has a total of eight controller ports (four on each controller), all I/O seems to be routed through just two ports on Controller 1. Note that SAN zoning is only allowing each HBA to see ports 1 and 2 of each controller, explaining why no I/O is seen on ports 3 and 4 even though round robin I/O path policy is being used.

The system is unbalanced because, despite having three Vdisks preferred to Controller 2, most of the workload is handled by Controller 1.

You can verify this imbalance by reviewing the appropriate Vdisk path information. Figure C-3 provides path information for Vdisk9; Figure C-4 provides information for Vdisk5.

Figure C-3. Path information for Vdisk9

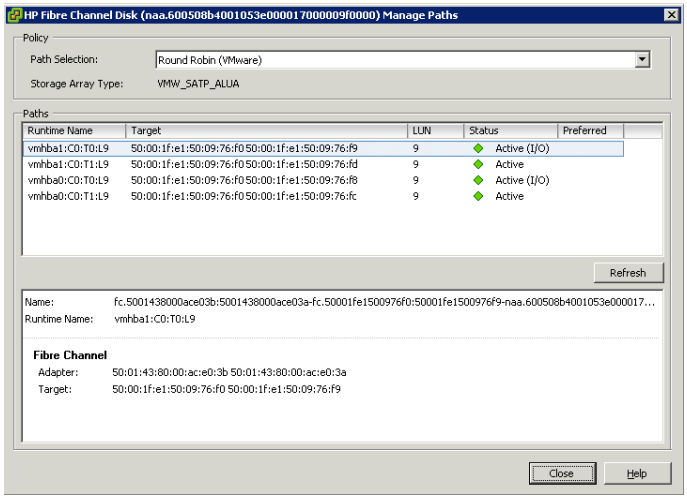
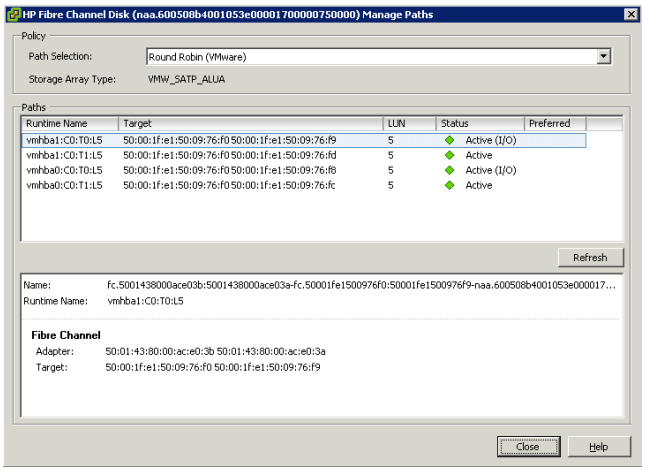


Figure C-4. Path information for Vdisk5



Alternatively, you can review Vdisk properties in Command View EVA to determine controller ownership, as shown in Figure C-5 (Vdisk9) and C-6 (Vdisk5).

Figure C-5. Vdisk properties for Vdisk9

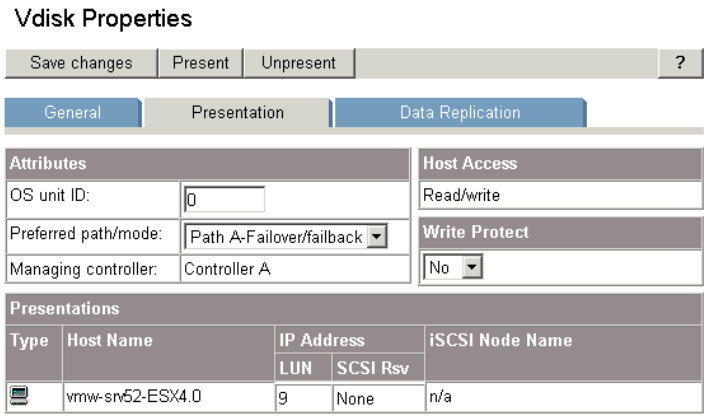
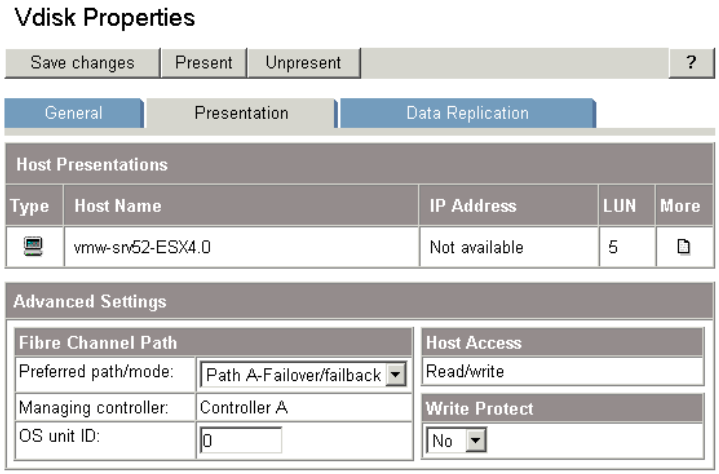


Figure C-6. Vdisk properties for Vdisk5



For a more granular view of throughput distribution, use the following command:

```
evaperf vd -sz <array_name> -cont X -dur Y
```

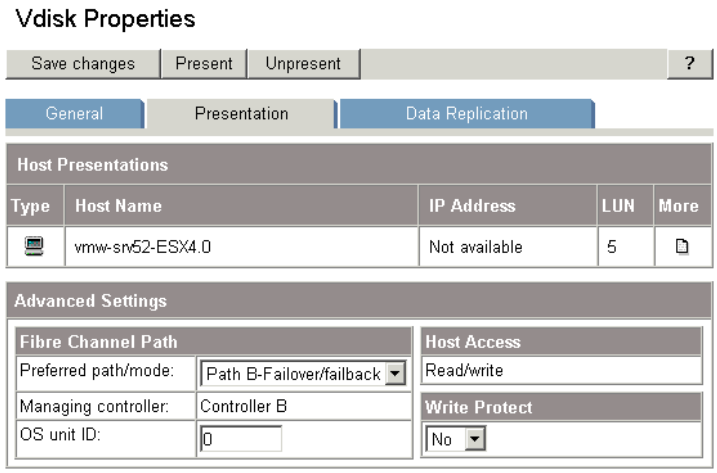
This command displays statistics at the EVA Vdisk level, making it easier for you to choose the appropriate Vdisk(s) to move from one controller to the other in order to better balance controller throughputs.

Moving the chosen Vdisk from one controller to the other

To better balance throughput in this example, Vdisk5 is being moved to Controller 2.

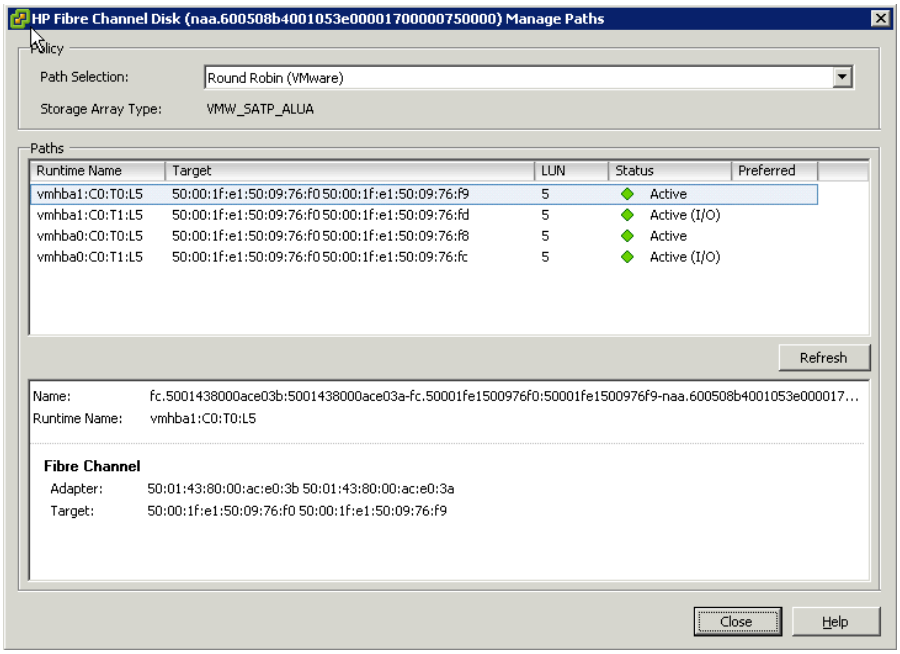
This move is accomplished by using Command View EVA to change the managing controller for Vdisk5, as shown in Figure C-7.

Figure C-7. Using Command View EVA to change the managing controller for Vdisk5 from Controller A to Controller B



After a rescan or vCenter refresh, you can verify that the change has been implemented, as shown in Figure C-8.

Figure C-8. Confirming that ownership has changed



I/O is now round robin on FP1 and FP2 of Controller B.

Validating the better balanced configuration

You can review the output of EVAperf (as shown in Figure C-9) to verify that controller throughput is now better balanced. Run the following command:

```
evaperf hps -sz <array_name> -cont X -dur Y
```

Figure C-9. Improved I/O distribution

EVAPerf - evaperf hps -sz EVA-DC2 -cont 2 -dur 66666666666

Name	Read Req/s	Read MB/s	Read Latency (ms)	Write Req/s	Write MB/s	Write Latency (ms)	Av. Queue Depth	Port WWN	Ctlr	Node
FP1	0	0.00	0.0	857	55.68	30.8	15	5000-1FE1-5009-76FC	008R	EVA-DC2
FP2	0	0.00	0.0	854	56.58	31.0	15	5000-1FE1-5009-76FD	008R	EVA-DC2
FP3	0	0.00	0.0	0	0.00	0.0	0	5000-1FE1-5009-76FE	008R	EVA-DC2
FP4	0	0.00	0.0	0	0.00	0.0	0	5000-1FE1-5009-76FF	008R	EVA-DC2
FP1	0	0.00	0.0	169	57.75	90.3	15	5000-1FE1-5009-76F8	Z07Y	EVA-DC2
FP2	0	0.00	0.0	181	65.18	89.8	15	5000-1FE1-5009-76F9	Z07Y	EVA-DC2
FP3	0	0.00	0.0	0	0.00	0.0	0	5000-1FE1-5009-76FA	Z07Y	EVA-DC2
FP4	0	0.00	0.0	0	0.00	0.0	0	5000-1FE1-5009-76FB	Z07Y	EVA-DC2

The system now has much better I/O distribution.

Appendix D: Caveat for data-in-place upgrades and Continuous Access EVA

The vSphere datastore may become invisible after one of the following actions:

- Performing a data-in-place upgrade from one EVA controller model to another
- Using Continuous Access EVA to replicate from one EVA model to another

Following these actions, ESX treats the new datastore as being a snapshot and, by default, does not display it.

Why is the datastore treated as a snapshot?

When building the VMFS file system on a logical unit, ESX writes metadata to the Logical Volume Manager (LVM) header that includes the following information:

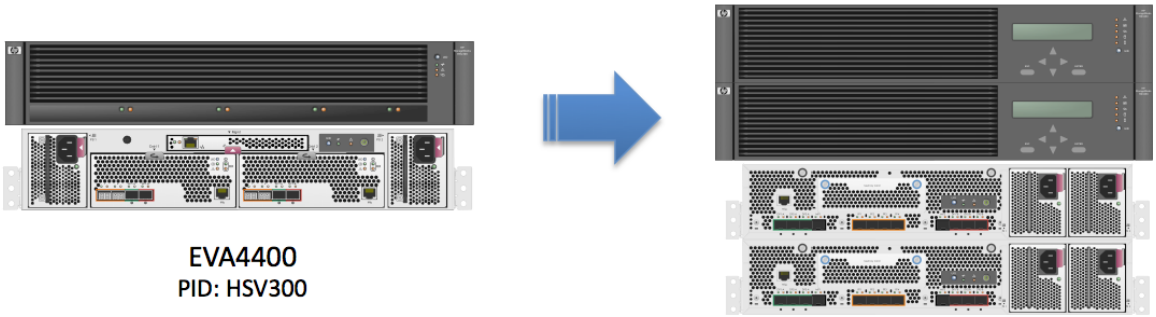
- Vdisk ID (such as Vdisk 1)
- SCSI inquiry string for the storage (such as HSV300); also known as the product ID (PID) or model string
- Unique Network Address Authority (NAA)-type Vdisk identifier, also known as the Worldwide Node LUN ID of the Vdisk

If any of these attributes changes after you create the new datastore, ESX treats the volume as a snapshot because the new Vdisk information will not match the metadata written on disk.

Example

Consider the data-in-place migration example shown in Figure D-1, where existing HSV300 controllers are being replaced with HSV450 controllers.

Figure D-1. Replacing EVAs and controllers



After the upgrade, all Vdisks will return "HSV450" instead of "HSV300" in the standard inquiry page response. This change in PID creates a mismatch between LVM header metadata and the information coming from the Vdisk.

Note
A similar mismatch would occur if you attempted to use Continuous Access EVA to replicate from the EVA4400 to the EVA8400.

When such a mismatch occurs, datastores are treated as snapshots and are not exposed to ESX. However, vSphere 4.x allows you to force-mount or re-signature these snapshots to make them accessible. For more information, refer to the following VMware Knowledge Base (KB) articles: 1011385 and 1011387.

Appendix E: Configuring VMDirectPath I/O for Command View EVA in a VM

This appendix describes how to configure VMDirectPath I/O in a vSphere environment for use with Command View EVA. An example is presented.

Note
The configuration described in this appendix is only provided for the purposes of this example.

Sample configuration

Server configuration

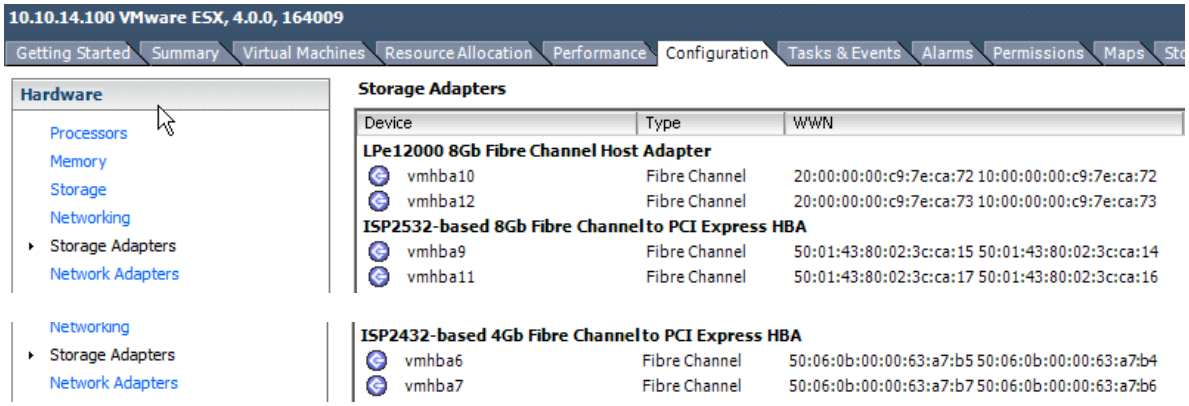
Table E-1 summarizes the configuration of the vSphere server used in this example.

Table E-1. vSphere server configuration summary

Component	Description	
ESX version	ESX 4.0 Build 164009	
Virtual machine	VM2 (Windows Server 2008)	
Local datastore	Storage 1	
HBA	HBA1 (Q4GB)	QLogic Dual Channel 4 Gb HBA
		Port 1: 5006-0B00-0063-A7B4
		Port 2: 5006-0B00-0063-A7B6
	HBA2 (Q8GB)	QLogic Dual Channel 8 Gb HBA
		Port 1: 5001-4380-023C-CA14
		Port 2: 5001-4380-023C-CA16
	HBA3 (E8GB)	Emulex Dual Channel 8 Gb HBA
		Port 1: 1000-0000-C97E-CA72
		Port 2: 1000-0000-C97E-CA73

By default, vSphere 4.x claims all HBAs installed in the system, as shown in the vSphere Client view presented in Figure E-1.

Figure E-1. Storage Adapters view, available under the Configuration tab of vSphere Client



This appendix shows how to assign HBA3 to VM2 in vSphere 4.x.

EVA configuration

This example uses four ports on an EVA8100 array (Ports 1 and 2 on each controller). A single EVA disk group was created.

The EVA configuration is summarized in Table E-2.

Table E-2. EVA array configuration summary

Component	**Description**		
EVA disk group	Default disk group, with 13 physical disks		
Vdisks	\VMDirectPath\ESX-VMFS-LUN1: 50GB	ESX LUN 1	Path A Failover/Failback
	\VMDirectPath\ESX-VMFS-LUN1: 50GB	ESX LUN 2	Path B Failover/Failback
	\VMDirectPath\ESX-VM-RDM-Win2k8: 40GB	ESX LUN 3 WIN VM: disk1 (RDM)	Path A Failover/Failback
	\VM-DirectLUNs\Win2k8-VM-dLUN1: 30GB	WIN LUN1	Path A Failover/Failback
	\VM-DirectLUNs\Win2k8-VM-dLUN2: 30GB	WIN LUN2	Path B Failover/Failback
Vdisk presentation	**vSphere server**		
	HBA1		
	Port 1: 5006-0B00-0063-A7B4		
	Port 2: 5006-0B00-0063-A7B6		
	Vdisks		
	\VMDirectPath\ESX-VMFS-LUN1: 50GB		
	\VMDirectPath\ESX-VMFS-LUN1: 50GB		
	\VMDirectPath\ESX-VM-RDM-Win2k8: 40GB		
	\VMDirectPath\ESX-VM-RDM-RHEL5: 40GB		

Component	Description	
	VM2 (Windows Server 2008 VM)	
	HBA3	
	Port 1: 1000-0000-C97E-CA72	
	Port 2: 1000-0000-C97E-CA73	
	Vdisks	
	\VM-DirectLUNs\Win2k8-VM-dLUN1: 30GB	
	\VM-DirectLUNs\Win2k8-VM-dLUN2: 30GB	
Host modes	**vSphere server**	VMware
	VM2	Windows Server 2008

Fibre Channel configuration

This example uses two HP 4/64 SAN switches, with a zone created on each.

The Fibre Channel configuration is summarized in Table E-3.

Table E-3. Fibre Channel configuration summary

Component	Description	WWN
Switch 1, Zone 1	Controller 1, Port 1	5000-1FE1-0027-07F8
	Controller 2, Port 1	5000-1FE1-0027-07FC
	HBA 1, Port 1	5006-0B00-0063-A7B4
	VM2, HBA3, Port 1	1000-0000-C97E-CA72
Switch 2, Zone 1	Controller 1, Port 1	5000-1FE1-0027-07F8
	Controller 2, Port 1	5000-1FE1-0027-07FC
	HBA 1, Port 1	5006-0B00-0063-A7B4
	VM2, HBA3, Port 1	1000-0000-C97E-CA72

Configuring the vSphere host

After the SAN topology and array-side configuration have been completed, you can configure HBA3 to be used as a VMDirectPath HBA for the Windows Server 2008 VM.

Note

If desired, you could configure VMDirectPath HBAs before configuring the SAN.

This appendix outlines a procedure for configuring VMDirectPath[16].

First, complete the following prerequisites:

- Open a PuTTY (ssh client) session[17] to the particular vSphere host.
- Open a vSphere Client connection to the vSphere host.

[16] This procedure assumes that you have never performed this task before. Alternate methods are available.
[17] While not necessary, an ssh session may be useful the first time you perform this procedure.

- Pre-install the VMs (for example, as VMs installed on a VMDK on a SAN datastore or a local datastore).

Note
Refer to Configuring EVA arrays for more information on placing VMs.

The procedure is as follows:

1. Identify which HBAs are present on the vSphere server by issuing the following command:

   ```
   [root@lx100 ~]# lspci | grep "Fibre Channel"
   ```

   This command provides a quick view of the HBAs in your system and their respective PCI hardware IDs. Alternatively, you can view HBAs via the vSphere Client; however, PCI hardware IDs would not be shown.

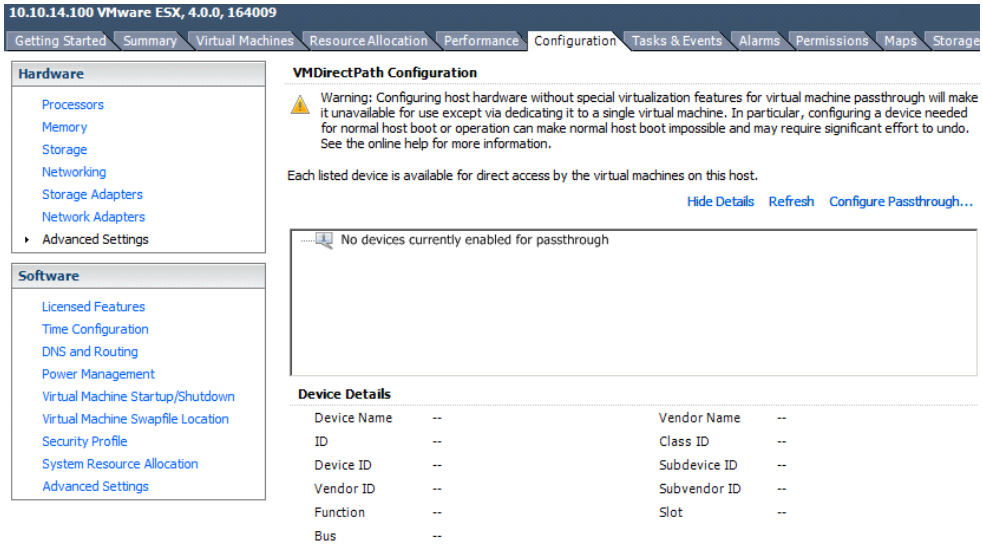
   The output to the above command is similar to that shown in Figure E-2.

Figure E-2. HBAs present on the vSphere server

```
10:00.0 Fibre Channel: QLogic Corp ISP2432-based 4Gb Fibre Channel to PCI Express HBA (rev 02)
10:00.1 Fibre Channel: QLogic Corp ISP2432-based 4Gb Fibre Channel to PCI Express HBA (rev 02)
1b:00.0 Fibre Channel: QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA (rev 02)
1b:00.1 Fibre Channel: QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA (rev 02)
21:00.0 Fibre Channel: Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter (rev 03)
21:00.1 Fibre Channel: Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter (rev 03)
```

2. Access the vSphere host through vSphere Client. Select the **Configuration** tab and click on **Advanced Settings** in the **Hardware** section, as shown in Figure E-3, to determine if passthrough (VMDirectPath) is supported.

Figure E-3. Indicating that no devices have been enabled for VMDirectPath

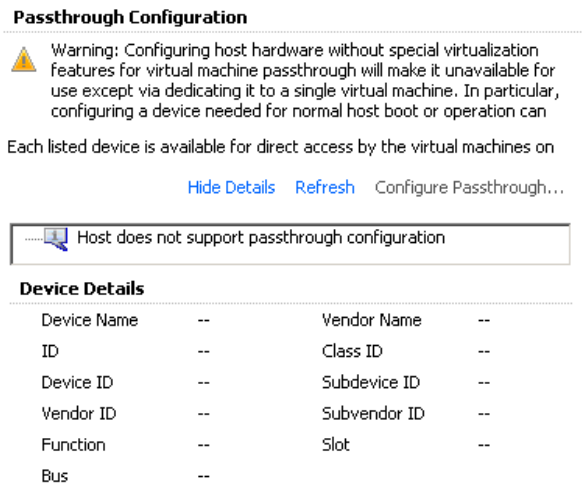


The screen displays a warning indicating that configuring a device for VMDirectPath will render that device unusable by vSphere.

In this example, no devices are currently enabled for VMDirectPath I/O.

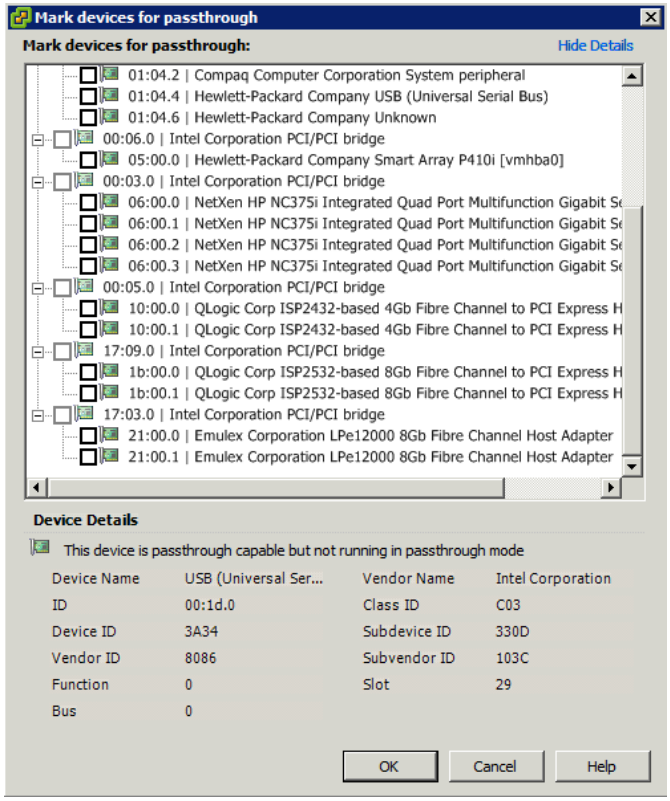
However, if your server hardware does not support Intel® Virtualization Technology for Directed I/O (VT-d) or AMD Extended Page Tables (EPT), Nested Page Tables (NPT), and Rapid Virtualization Indexing (RVI), it cannot support VMDirectPath. In this case, the **Advanced Settings** screen would be similar to that shown in Figure E-4, which indicates that the host does not support VMDirectPath.

Figure E-4. Indicating that, in this case, the server hardware is incompatible and that VMDirectPath cannot be enabled


Passthrough Configuration
Warning: Configuring host hardware without special virtualization features for virtual machine passthrough will make it unavailable for use except via dedicating it to a single virtual machine. In particular, configuring a device needed for normal host boot or operation can
Each listed device is available for direct access by the virtual machines on
Hide Details
Refresh
Configure Passthrough...
Host does not support passthrough configuration
Device Details
Device Name -- Vendor Name --
ID -- Class ID --
Device ID -- Subdevice ID --
Vendor ID -- Subvendor ID --
Function -- Slot --
Bus --

3. If your server has compatible hardware, click on the **Configure Passthrough...** link to move to the **Mark devices for passthrough** page, as shown in Figure E-5. Review the device icons:
   - **Green**: Indicates that the device is passthrough-capable but not currently running in passthrough mode
   - **Orange arrow**: Indicates that the state of the device has changed and that the server needs to be rebooted for the change to take effect

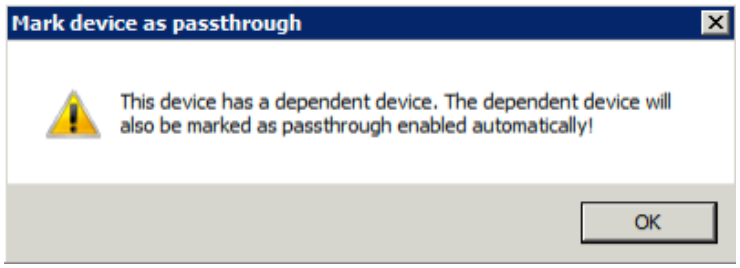
Figure E-5. Allowing you to select VMDirectPath on the desired device(s)


Mark devices for passthrough
Mark devices for passthrough:
Hide Details
01:04.2 | Compaq Computer Corporation System peripheral
01:04.4 | Hewlett-Packard Company USB (Universal Serial Bus)
01:04.6 | Hewlett-Packard Company Unknown
00:06.0 | Intel Corporation PCI/PCI bridge
05:00.0 | Hewlett-Packard Company Smart Array P410i [vmhba0]
00:03.0 | Intel Corporation PCI/PCI bridge
06:00.0 | NetXen HP NC375i Integrated Quad Port Multifunction Gigabit Se
06:00.1 | NetXen HP NC375i Integrated Quad Port Multifunction Gigabit Se
06:00.2 | NetXen HP NC375i Integrated Quad Port Multifunction Gigabit Se
06:00.3 | NetXen HP NC375i Integrated Quad Port Multifunction Gigabit Se
00:05.0 | Intel Corporation PCI/PCI bridge
10:00.0 | QLogic Corp ISP2432-based 4Gb Fibre Channel to PCI Express H
10:00.1 | QLogic Corp ISP2432-based 4Gb Fibre Channel to PCI Express H
17:09.0 | Intel Corporation PCI/PCI bridge
1b:00.0 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express H
1b:00.1 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express H
17:03.0 | Intel Corporation PCI/PCI bridge
21:00.0 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
21:00.1 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
Device Details
This device is passthrough capable but not running in passthrough mode
Device Name USB (Universal Ser... Vendor Name Intel Corporation
ID 00:1d.0 Class ID C03
Device ID 3A34 Subdevice ID 330D
Vendor ID 8086 Subvendor ID 103C
Function 0 Slot 29
Bus 0
OK
Cancel
Help

4. Select the desired devices for VMDirectPath; select and accept the passthrough device dependency check shown in Figure E-6.

IMPORTANT
If you select **OK**, the dependent device is also configured for VMDirectPath, regardless of whether or not it was being used by ESX.
If your server is booting from SAN, be careful not to select the incorrect HBA; your server may subsequently fail to reboot.

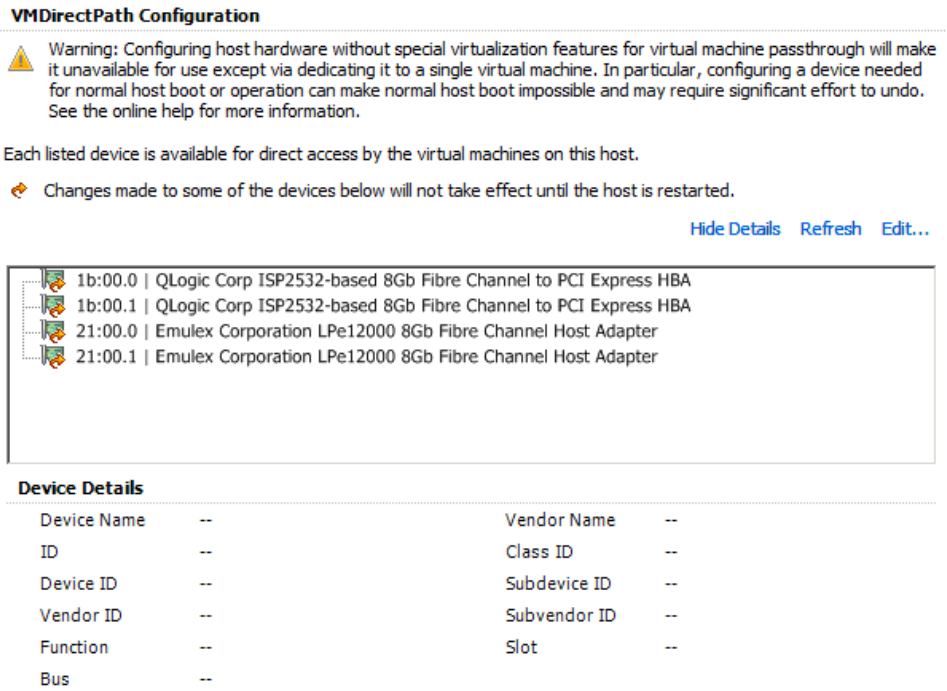
Figure E-6. Warning about device-dependency


Mark device as passthrough
This device has a dependent device. The dependent device will also be marked as passthrough enabled automatically!
OK

As shown in Figure E-7, the VMDirectPath Configuration screen reflects the changes you have made.

Device icons indicate that the changes will only take effect when the server is rebooted.

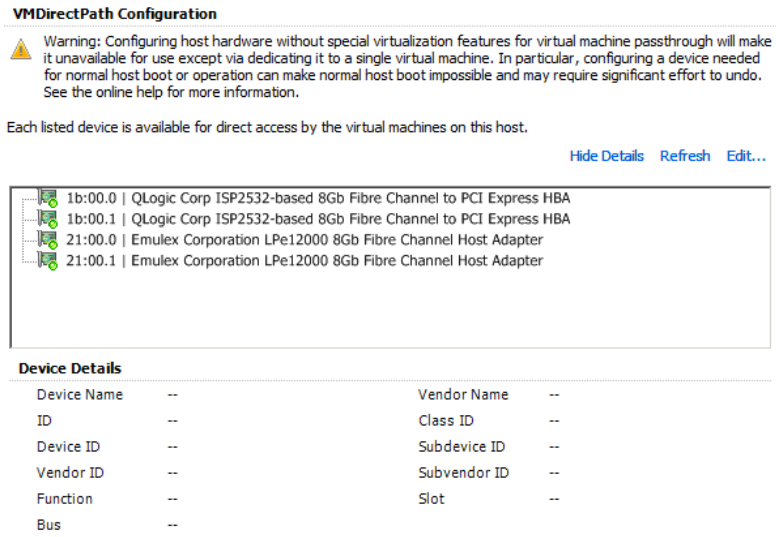
Figure E-7. Indicating that four HBA ports have been enabled for VMDirectPath but that these changes will not take effect until a server reboot


VMDirectPath Configuration
Warning: Configuring host hardware without special virtualization features for virtual machine passthrough will make it unavailable for use except via dedicating it to a single virtual machine. In particular, configuring a device needed for normal host boot or operation can make normal host boot impossible and may require significant effort to undo. See the online help for more information.
Each listed device is available for direct access by the virtual machines on this host.
Changes made to some of the devices below will not take effect until the host is restarted.
Hide Details
Refresh
Edit...
1b:00.0 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
1b:00.1 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
21:00.0 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
21:00.1 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
Device Details
Device Name --
ID --
Device ID --
Vendor ID --
Function --
Bus --
Vendor Name --
Class ID --
Subdevice ID --
Subvendor ID --
Slot --

5. Reboot the server through the vSphere client or the command line.

6. After the reboot, confirm that device icons are green, as shown in Figure E-8, indicating that the VMDirectPath-enabled HBA ports are ready to use.

Figure E-8. The HBA ports have been enabled for VMDirectPath and are ready for use


VMDirectPath Configuration
Warning: Configuring host hardware without special virtualization features for virtual machine passthrough will make it unavailable for use except via dedicating it to a single virtual machine. In particular, configuring a device needed for normal host boot or operation can make normal host boot impossible and may require significant effort to undo. See the online help for more information.
Each listed device is available for direct access by the virtual machines on this host.
Hide Details
Refresh
Edit...
1b:00.0 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
1b:00.1 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
21:00.0 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
21:00.1 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
Device Details
Device Name -- Vendor Name --
ID -- Class ID --
Device ID -- Subdevice ID --
Vendor ID -- Subvendor ID --
Function -- Slot --
Bus --

7. Issue the following command to validate the VMDirectPath-enabled HBA ports:

```
[root@lx100 ~]# vmkchdev -l | grep vmhba
```

Review the resulting output, which is shown in Figure E-9.

Figure E-9. Validating that four HBA ports have indeed been enabled for VMDirectPath

```
000:31.2 8086:3a20 103c:330d vmkernel vmhba1
005:00.0 103c:323a 103c:3245 vmkernel vmhba0
016:00.0 1077:2432 103c:7041 vmkernel vmhba6
016:00.1 1077:2432 103c:7041 vmkernel vmhba7
027:00.0 1077:2532 103c:3263 passthru vmhba9
027:00.1 1077:2532 103c:3263 passthru vmhba11
033:00.0 10df:f100 103c:3282 passthru vmhba10
033:00.1 10df:f100 103c:3282 passthru vmhba12
```

As expected, the following devices have been enabled for VMDirectPath and are no longer claimed by the VMkernel.

- Hardware ID 1b:00.0/1b:00.1 (hexadecimal), 027:00.0/027:00.1 (decimal)
- Hardware ID 21:00.0/21:00.1 (hexadecimal), 033:00.0/033:00.1 (decimal)

Furthermore, the vSphere Client **Storage Adapters** window no longer displays vmhba9, 10, 11, and 12, as shown in Figures E-1 and E-2. The VMDirectPath HBAs can now be assigned to VMs.

Note
The changes you have just made are stored in the **/etc/vmware/esx.conf** file.

Configuring the array

Use Command View EVA to perform the following steps:

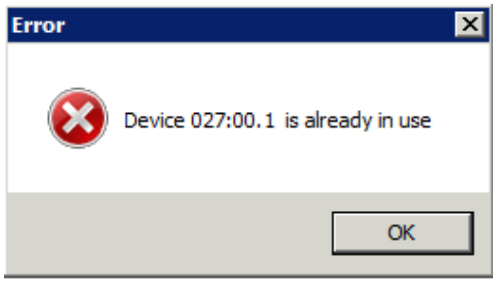
1. Create the Vdisks.
2. Add the hosts:
   - vSphere server: Set the Command View EVA host mode to VMware
   - VM2: Set the Command View EVA host mode to Windows Server 2008
3. Add Vdisks presentation.

Configuring the VM

Caveats

- HBA ports are assigned to the VM one at a time, while the VM is powered off.
- The VM must have a memory reservation for the fully-configured memory size.
- You must not assign ports on the same HBA to different VMs, or the same HBA to various VMs. Though such configurations are not specifically prohibited by vSphere client, they would result in the VM failing to power on. You would receive a message such as that shown in Figure E-10.

Figure E-10. Message resulting from a misconfiguration



Prerequisites

Before beginning the configuration, complete the following prerequisites:

- Open a vSphere Client connection to the vSphere host.
- Pre-install the VM (for example, on a VMDK on a local or SAN datastore).
- Obtain console access to the VM through vSphere Client.

Note

Refer to Configuring EVA arrays for more information on placing VMs.

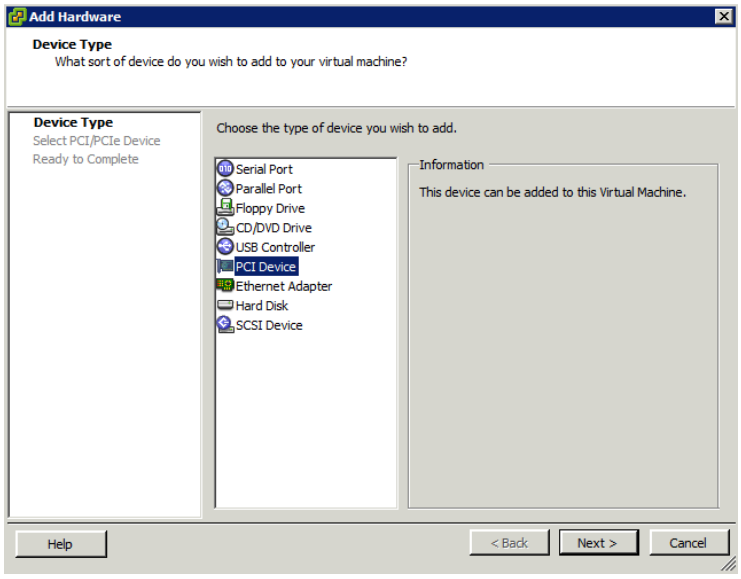
Procedure

Carry out the following steps to add VMDirectPath devices to a selected VM:

1. From the vSphere client, select VM2 from the inventory, ensuring that it is powered off.
2. Right-click on the VM and select **Edit Settings**.
3. Select the **Hardware** tab and then click on **Add**.

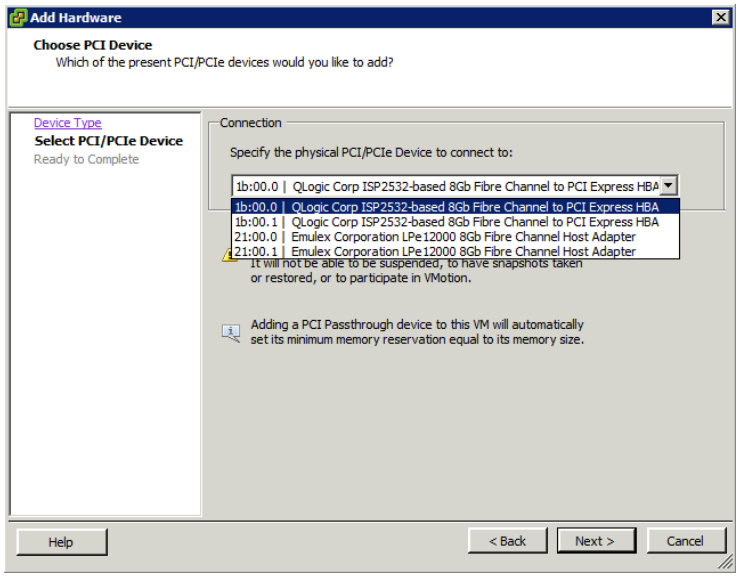
4. Select **PCI Device** and then click on **Next**, as shown in Figure E-11.

Figure E-11. Selecting **PCI Device** as the type of device to be added to the VM



5. From the list of VMDirectPath devices, select the desired device to assign to the VM, as shown in Figure E-12. In the example, select Port 1 of HBA3 (that is, device 21:00.0).
   For more information on selecting devices, refer to Caveats.

Figure E-12. Selecting VMDirectPath devices to be added to the VM


Add Hardware
Choose PCI Device
Which of the present PCI/PCIe devices would you like to add?
Device Type
Select PCI/PCIe Device
Ready to Complete
Connection
Specify the physical PCI/PCIe Device to connect to:
1b:00.0 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
1b:00.0 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
1b:00.1 | QLogic Corp ISP2532-based 8Gb Fibre Channel to PCI Express HBA
21:00.0 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
21:00.1 | Emulex Corporation LPe12000 8Gb Fibre Channel Host Adapter
or restored, or to participate in VMotion.
Adding a PCI Passthrough device to this VM will automatically set its minimum memory reservation equal to its memory size.
Help
< Back
Next >
Cancel

6. Repeat Step 5 to assign Port 2 of HBA2 (that is, device 21:00.1) to the VM.
7. Use vSphere Client to open a console window to the Windows Server 2008 VM.
8. Use Device Manager on the VM to verify that the Emulex HBA has been assigned to this VM.

If zoning has already been implemented (see Fibre Channel configuration), you can now follow the HP Command View EVA installation guide to install Command View EVA, as you would on a bare-metal (physical) server.

For more information

Data storage from HP	hp.com/go/storage
HP and VMware	hp.com/go/vmware
Converged Storage for VMware	http://www8.hp.com/us/en/products/data-storage/data-storage-products.html?compURI=1285027
Documentation for EVA arrays	http://h20566.www2.hp.com/portal/site/hpsc/public/psi/manualsResults?sp4ts.oid=5062117&ac.admitted=1394743561397.876444892.199480143
HP Command View EVA installation guide	http://h20000.www2.hp.com/bizsupport/TechSupport/DocumentIndex.jsp?contentType=SupportManual&lang=en&cc=us&docIndexId=64179&taskId=125&prodTypeId=18964&prodSeriesId=5061965
Product documentation for HP Insight Control for VMware vCenter Server	http://h18004.www1.hp.com/products/servers/management/integration.html

To help us improve our documents, please provide feedback at hp.com/solutions/feedback.

Sign up for updates
hp.com/go/getupdated

© Copyright 2009, 2011, 2012, 2013, 2014 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft, Windows, and Windows Vista are U.S. registered trademarks of Microsoft Corporation. AMD is a trademark of Advanced Micro Devices, Inc. Intel is a trademark of Intel Corporation in the U.S. and other countries.

4AA1-2185ENW, March 2014, Rev. 7